

# A Many-Objective Evolutionary Algorithm based on Rotation and Decomposition

Juan Zou<sup>a,b</sup>, Jing Liu<sup>a,b,\*</sup>, Shengxiang Yang<sup>a,d,\*</sup>, Jinhua Zheng<sup>a,b,c</sup>

<sup>a</sup>*Key Laboratory of Hunan Province for Internet of Things and Information Security, Xiangtan University, Xiangtan, Hunan, 411105, China.*

<sup>b</sup>*Key Laboratory of Intelligent Computing and Information Processing, Ministry of Education, Information Engineering College of Xiangtan University, Xiangtan, Hunan, 411105, China.*

<sup>c</sup>*Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang, 421002, China*

<sup>d</sup>*School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K.*

---

## Abstract

Evolutionary algorithms have shown their promise in addressing multi-objective problems (MOPs). However, the Pareto dominance used in multi-objective optimization loses its effectiveness when addressing many-objective problems (MaOPs), which are defined as having more than three objectives. This is because the Pareto dominance loses its ability to distinguish between individuals. In this paper, a many-objective evolutionary algorithm based on rotation and decomposition is proposed (MaOEA-RD) to overcome the shortcoming of insufficient selection pressure caused by the Pareto dominance. First, the coordinates system is rotated and a hyperplane is established to distinguish between the nondominated individuals. Then, a novel individual selection mechanism incorporating decomposition is adopted to maintain the diversity of the population. In order to compensate for the deficiency of the predefined reference vectors, a reference vector adjustment mechanism is proposed. Experimental studies on several well-known benchmark problems show that the proposed algorithm is competitive compared with nine state-of-the-art many-objective algorithms.

---

\*Corresponding author: Jing Liu, Shengxiang Yang

*Email addresses:* liujinghn@qq.com (Jing Liu), syang@dmu.ac.uk (Shengxiang Yang)

*Keywords:* Many-objective optimization, evolutionary algorithm, Pareto optimality, rotated coordinates, decomposition.

---

## 1. Introduction

Multiobjective optimization problems (MOPs) can be mathematically defined as [1]:

$$\begin{cases} \min F(X) = (f_1(X), f_2(X), \dots, f_m(X)), \\ \text{subject to } X \in \Omega, \end{cases} \quad (1)$$

where  $X = (x_1, x_2, \dots, x_n)$  is an  $n$ -dimensional decision variable vector from the decision space  $\Omega$ ;  $F(X)$  is an objective function vector that consists of  $m$  conflicting objective functions.

Unlike single-objective optimization, a set of trade-off solutions, termed Pareto optimal solutions, is expected to be found for MOPs. Let  $X_1, X_2 \in \Omega$ ;  $X_1$  is said to dominate  $X_2$ , denoted by  $X_1 \prec X_2$ , if and only if  $f_i(X_1) \leq f_i(X_2)$  for each  $i \in \{1, \dots, m\}$  and  $f_j(X_1) < f_j(X_2)$  for at least one index  $j \in \{1, \dots, m\}$ ; if all  $X$  from  $\Omega$  cannot dominate  $X_1$ , we call  $X_1$  a nondominated or Pareto optimal solution. The set of all Pareto optimal solutions is called the Pareto optimal set (PS) and the set of all the Pareto optimal objective vectors is called the Pareto optimal front (PF). Any improvement in one objective of a Pareto optimal solution is bound to deteriorate at least one other objective.

Multiobjective optimization algorithms, such as NSGA-II [2] and SPEA-II [3], perform excellently with MOPs but performance greatly drops in dealing with many-objective optimization problems (MaOPs), which have four or more objectives [4]. The main reason is that the Pareto dominance loses its ability to distinguish between individuals [5]. To overcome the drawback of Pareto-dominance-based multiobjective evolutionary algorithms (MOEAs) in addressing MaOPs, many efforts have been made. These can be mainly summarized as the following categories.

The first category is the development of new dominance relations that can increase the selection pressure.  $\varepsilon$ -dominance [6] makes the nondominated individuals distinguishable by introducing the parameter  $\varepsilon$  that expands the Pareto dominated region of the individual. *Grid-dominance* [7] divides the objective space into several grids and distinguishes individuals by grid coordinates instead of the Pareto dominance relation. *SDR dominance* [8]

balances proximity and diversity by developing an adaptive niching technique based on the angles between the candidate solutions and maintains only one candidate solution with the best convergence in each niche. *GWS-PLS* [9] proposes the grid weighted sum dominance by combining the Pareto dominance and weighted sum approach in a grid system to address combinatorial multiobjective optimization problems. In addition, *Rotated-grid* [10], *RP-dominance* [11],  *$\theta$ -dominance* [12], *fuzzy-based Pareto dominance* [13] and *rank-dominance* [14] have demonstrated their abilities for handling MaOPs.

The second category includes the non-Pareto-dominance-based methods, which are mainly decomposition-based and indicator-based. MOEA/D [15] represents the decomposition framework. In the decomposition-based category, the objective space is decomposed into a set of subproblems by uniformly distributed reference vectors, and all subproblems are optimized in a collaborative manner. Several algorithms have been designed, such as NSGA-III [16] and MOEA/DD [17] as well as others [18, 19, 20, 21, 22, 23]. The indicator is an important reference to evaluate the performance of an algorithm. There are several indicators, such as diversity indicator [24, 25, 26], convergence indicator [27] and comprehensive performance indicator [28, 29]. HyPE [30], MOMBI [31] and MaOEA/IGD [32], which are indicator-based algorithms, map the proximity and diversity of the population into designated indicators. The excellent performance of indicator-based algorithms has attracted the attention of many scholars, although the extensive computation required by some indicators in high-dimensional objective space still remains a problem.

The last category of algorithms adopts multiple archives and a new density management mechanism. Two.arch2 [33] adopts two archives, one for storing individuals with excellent convergence and the other for storing individuals with excellent distribution. 1by1EA [34] uses a one-by-one selection strategy, which uses three archives to store individuals temporarily. SDE [35] is a strategy of density management that makes the density of solutions with poor convergence increase after transferring, while the density of individuals with excellent convergence has little influence. Some recent works focusing on the hybridization of the aforementioned categories have also achieved good results [36, 37, 38].

To test the performance of evolutionary algorithms, a series of benchmark instances have been proposed. Deb et al. [39] suggested three different approaches for systematically designing test problems. The simplicity of construction, scalability to any number of decision variables and objectives,

knowledge of the shape and the location of the resulting PF, and introduction of controlled difficulties in both converging to the true PF and maintaining a widely distributed set of solutions are the main features of the DTLZ problems. Huband et al. [40] introduced a set of test problem criteria, which were in turn supported by a set of definitions, and presented a flexible toolkit for constructing well-designed test problems. Cheng et al. [41] carefully selected and modified 15 test problems with diverse properties to construct a benchmark test suite, aiming to promote the research of evolutionary many-objective optimization by suggesting a set of test problems with a good representation of various real-world scenarios. Just recently, a set of ten new test problems, with objective scalability, complicated PS, bias, disconnection, and degeneracy, were constructed by Li et al. [42]. Among them, DTLZ and WFG are two representative benchmark suites with scalability in the number of variables and objectives.

The Pareto dominance loses its effectiveness in MaOPs, which makes the mission to select outstanding individuals almost depend on the mechanism of maintaining diversity. In this paper, a many-objective optimization algorithm based on rotation and decomposition (MaOEA-RD) is proposed. We use the *CI* proposed by Shen et al. [43] to enhance selection pressure on individuals and adopt the decomposition method to maintain diversity. The contributions of this paper are summarized as follows:

- 1) A novel individual selection mechanism incorporating decomposition is adopted to enhance the selection pressure on individuals.
- 2) To make up the deficiency of the predefined reference vectors, a reference vector adjustment mechanism is proposed.

The remainder of this paper is organized as follows: Section 2 introduces the background knowledge of this work. The details of MaOEA-RD are presented in Section 3. To evaluate the performance of our algorithm in addressing MaOPs, a series of experiments is presented and analyzed in Section 4. Finally, Section 5 concludes this paper.

## 2. Preliminaries

In this section, the ISNPS [43] proposed by Shen et al., which originally proposed the coordinates rotation, is briefly introduced first. Then, we introduce the decomposition of a MOP and the *PBI* approach. After that,

the representative decomposition-based algorithms with reference vector adjustment are summarized. Finally, the motivation of our paper is presented.

### 2.1. ISNPS

Each individual's behavior is separated into convergence information ( $CI$ ) and diversity information ( $DI$ ) in ISNPS. The core idea of ISNPS can be illustrated by Figure 1. Let  $f_1=(1,0)$  and  $f_2=(0,1)$  be the original coordinates' axes. As can be seen in Figure 1 (a), the first axis  $f_1$  is rotated to the  $r_1$ , which is paralleled with  $(1,1)$ . Thereafter, the second rotated axis  $r_2$  is generated through orthogonalizing  $r_1$ . ISNPS calculates the coordinates  $F_r(X) = (f_{r1}(X), f_{r2}(X))$  to the original objective vector  $F(X) = (f_1(X), f_2(X))$  on the rotated coordinates axes ( $r_1$  and  $r_2$ ), as presented in Figure 1 (b). The transformed coordinates  $r_1$  and  $r_2$  are defined as  $CI$  and  $DI$ , respectively. After the rotation operation, the convergence information of each individual can be better highlighted. ISNPS selects individuals with good convergence through the values of  $CI$ , and maintains the distribution of the population through the values of  $DI$ .

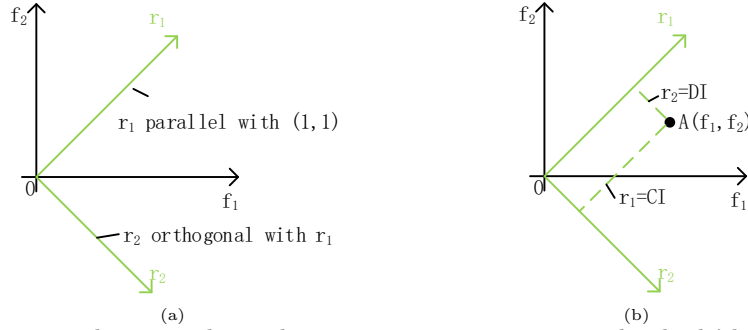


Figure 1: The rotated coordinates system to separate individuals' behaviors.

There is a series of selections within each evolution in ISNPS. In each selection, ISNPS selects the individual with the smallest  $CI$  value ( $X^*$ ) out of the entire population and ignores individuals within  $X^*$ 's neighborhood by using the  $DI$  value, which is constructed by two cones. The cones can be calculated by formula 2 and formula 3, respectively.

$$Cone_1(X^*) = \{X | \arctan \frac{\|DI(X^*) - DI(X)\|}{3 \times CI(X)} \leq \alpha\}, \quad (2)$$

$$Cone_2(X^*) = \{X | \arccos \frac{F(X^*) \cdot (F(X) - F(X^*))}{\|F(X^*)\| \|F(X) - F(X^*)\|} \leq 45^\circ\}, \quad (3)$$

where  $X^*$  is the selected individual;  $\alpha$  is a parameter of ISNPS and  $\|\cdots\|$  is the Euclidean distance. Further details of ISNPS can be found in [43].

In fact, the  $CI$  is the distance between the individual and the hyperplane perpendicular to the convergence direction vector  $(1, 1, \dots, 1)$ , which can largely reflect the convergence of an individual. The  $DI$  consists of the remaining objectives' values after removing the  $CI$ . Although the  $CI$  of the individual is removed, it is not appropriate for the  $DI$  to keep the population well distributed. In addition, the parameter  $\alpha$  in ISNPS has a great impact on the performance of the algorithm, and different values need to be set for different problems.

## 2.2. MOEA/D and PBI approach

In principle, the task of approximating the PF can be decomposed into several scalar optimizations, each of which is formulated as a weighted aggregation of all the individuals' objectives [15, 22]. MOEA/D pioneers the use of decomposition in evolutionary algorithms, whose key idea is to decompose a MOP into a number of single-objective optimization subproblems through aggregation. In MOEA/D, a set of uniformly spread reference vectors  $((W_1, \dots, W_i, \dots, W_N))$  is predefined, and each  $W_i ((w_1, w_2, \dots, w_m))$  satisfies  $\sum_{k=1}^m w_k = 1$  and  $\forall w_k \geq 0$ , where  $N$  is the size of population and  $m$  is the number of objectives. Generally, three aggregation functions, weighted sum, Tchebycheff and  $PBI$  function can well serve the purpose in MOEA/D. Take the  $PBI$  approach as an example. A MOP can be decomposed into  $N$  single-objective subproblems represented as formula 4,  $d_1$  and  $d_2$  can be computed, respectively, as formula 5 and formula 6, and  $Z^*$  is the ideal point<sup>1</sup>. In the original MOEA/D, an offspring solution is generated and compared with the solutions among some neighboring reference vectors. A parent solution can be replaced by an offspring only when it has a better aggregation function value. Further details of MOEA/D can be found in [15].

$$\begin{cases} \min g^{pbi}(X | W, Z^*) = d_1 + \theta d_2, \\ \text{subject to } X \in \Omega. \end{cases} \quad (4)$$

$$d_1 = \frac{W_i \cdot F(X_j)}{\|W_i\|}, \quad (5)$$

---

<sup>1</sup> $Z^* = (z_1, \dots, z_i, \dots, z_m)$ ,  $z_i = \min_{X \in \Omega} \{f_i(X)\}$ .

$$d_2 = \|d_1 \times \frac{W_i}{\|W_i\|} - F(X_j)\|. \quad (6)$$

Let's focus on the *PBI* aggregation function. Figure 2 presents the contours of the *PBI* function, with  $\theta=0$  (red),  $\theta=1$  (blue) and  $\theta=5$  (green), for reference vector  $W_i=(0.5, 0.5)$ . Different settings of  $\theta$  lead to distinct search behaviors of the *PBI* approach [15]. When the  $\theta$  is set to 0, the *PBI* approach is the same as the weighted sum approach. If the  $\theta$  is set to 1, the contours of the *PBI* function are the same as those of the Tchebycheff. Usually, the  $\theta$  is set to 5 for empirical [12, 17]. Figure 3 is an example when the  $\theta$  is set to 5 to explain the characteristics of the *PBI* approach. The  $g^{pbi}(X_i | W, Z^*)$  of individuals on the green, blue, and red contours increases successively. Specifically,  $g^{pbi}(X_B | W, Z^*) = g^{pbi}(X_D | W, Z^*) < g^{pbi}(X_C | W, Z^*) < g^{pbi}(X_A | W, Z^*)$ . From these four  $g^{pbi}$  values, we can observe that the *PBI* approach sacrifices convergence to ensure diversity. For this reason, the *PBI* approach has a probability of abandoning boundary solutions. For example,  $g^{pbi}(X_B | W, Z^*)$  is equal to  $g^{pbi}(X_D | W, Z^*)$ , which means solution B may not be selected. In addition, we exclude other factors and the evolutionary pressure of the *PBI* approach can be treated as gradients, as shown in Figure 4. The proximity term, which calculates the length of the projection of the individual on the nearest reference vector, guides the evolution of individuals towards the origin. As can be observed from Figure 5, the contours with the same *PBI* values fluctuate around concave concentric arcs, which may indicate that the *PBI* approach has a natural preference for concave PFs.

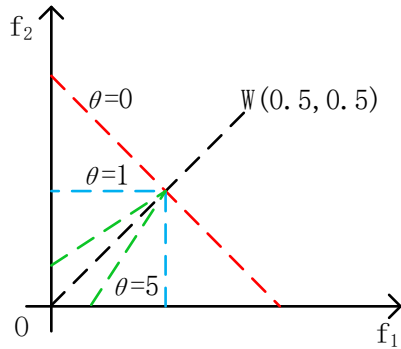


Figure 2: The contours of *PBI* function with different  $\theta$ s.

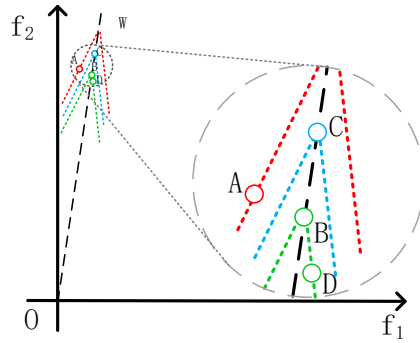


Figure 3: An example of *PBI* function with  $\theta=5$ .

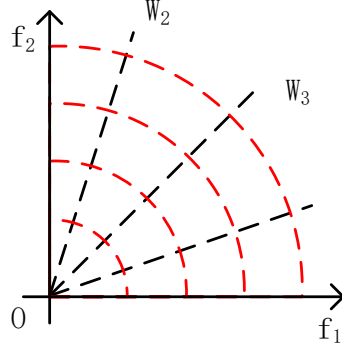


Figure 4: The proximity term of the *PBI* approach.

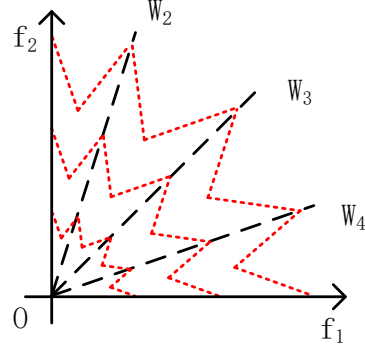


Figure 5: The contours with the same *PBI* values.

### 2.3. Algorithms with reference vector adjustment

Recent work in adapting reference vectors has attracted considerable research attention. This is because algorithms based on predefined uniformly distributed reference vectors cannot solve problems with irregular PFs (PFs, e.g., degenerate and disconnected ones) [44, 45, 46].

There have been several works on adjusting the reference vectors for MOEA/D. For example, Gu and Liu [47] adjusted the vectors using a linear interpolation of the nondominated solutions to approximate the PF. Jiang et al. [48] presented an adaptive vectors adjustment method which samples the regression curve of the reference vectors on the basis of an external archive. Gu et al. [49] used the equidistant interpolation to periodically update the reference vectors on the estimated PF. Jian and Deb [50] adjusted the reference vectors by simply deleting nonuseful ones and adding a simplex of  $m$  points centering around an existing reference vector. Qi et al. [51], a decomposition working population and an external archive, and the vectors were adaptively adjusted by estimating the sparsity of both working population and nondominated solutions in the external archive. Cheng et al. [52] adopted two reference vector adaptationsto deal with badly-scaled PFs and irregular PFs respectively. Cai et al. [53] proposed two stages of vector adjustments for many-objective problems, with one aiming at the number of the direction vectors and the other aiming at the positions of the direction vectors. Li et al. [54] sought a suitable distribution of weights for the given problem by elaborating five parts in the vectors adaptation. Farias et al. [55] used the uniformly random methodan approach to subproblems genera-



tion, and a flexible population size even when working with many-objective problems was allowed. Li et al. [56] established an archive to store the non-dominated solutions to reflect the PF, and these solutions were adopted to adapt the reference vectors.

Many researchers have proven that the method of adaptive adjustment of reference vectors effectively solves problems with irregular PFs [45, 46, 50, 51, 53, 57]. However, this method is counterproductive to problems with regular PFs. For a regular PF, there may exist multiple reference vectors corresponding to one single solution during the evolutionary process. As such, a change of the reference vectors which were already ideal for the considered regular PF may lead solutions towards wrong search directions [58].

#### 2.4. Motivation

From the previous analysis, we use the strength of  $CI$  as our first criterion to enhance the selection pressure. In order to reduce the preference of the  $PBI$  approach for concave PFs,  $d_1$  in the  $PBI$  is replaced by  $CI$  as our second criterion<sup>2</sup>. Therefore, the evolutionary direction of the  $PBI_m$  becomes parallel to the convergence direction vector  $(1,1,...,1)$ , as shown in Figure 6. The contours fluctuate around the parallel lines, as shown in Figure 7. In addition, these contours have a feature in which the closer the contour is to the coordinate axis, the smaller the angle between the contours and the reference vector, so that the boundary solutions can be better preserved. Although the  $PBI_m$  has a preference for linear PFs, it is a compromised solution with the expectation that it will perform versatilely on problems with diverse PF curvatures. In the experimental studies, we validate the versatility and efficiency of combining  $CI$  and  $PBI_m$  compared to  $PBI$ .

At the same time, in order to deal with MaOPs with regular or irregular PFs, we propose a reference vector adjustment mechanism, which combines with the  $CI$  to turn the reference vector adjustment operation on or off.

---

<sup>2</sup>In the rest of this paper, the modified  $PBI$  method is denoted as  $PBI_m$ .

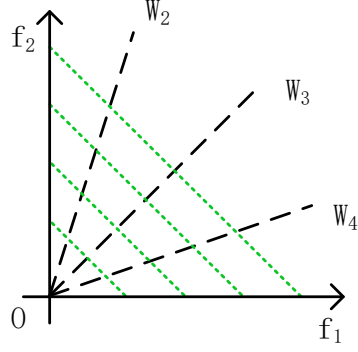


Figure 6: The proximity term of the  $PBI_m$  approach.

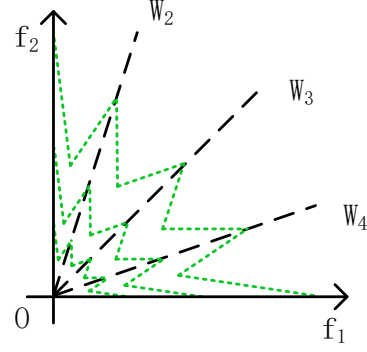


Figure 7: The contours with the same  $PBI_m$  values.

### 3. Proposed algorithm

#### 3.1. Framework

The framework of our algorithm is presented in Algorithm 1. MaOEA-RD starts by generating a set of uniformly distributed reference vectors  $W^3$  and an initial population  $P$  of size  $N$ . The  $HisCI$ , the  $HisW$  and the  $Flag$  is prepared for the reference vector adjustment mechanism (Algorithm 4); the  $t$  represents the number of iterations of the evolution of the population. The simulated binary crossover ( $SBX$ ) and the polynomial mutation [60] are employed to generate an offspring set  $Q_t$ . After merging  $P_t$  and  $Q_t$ , environmental selection (Algorithm 3) is carried out. It should be noted that MaOEA-RD will not adjust the reference vector at every generation by our reference vectors adjustment mechanism (Algorithm 4), the details are described later.

#### 3.2. Rotated coordinates

The number of nondominated individuals growing exponentially causes the non-convergence phenomenon of algorithms in high-dimensional space. The reason is that the Pareto dominance loses the ability to identify individuals. Therefore, we use the strength of  $CI$  proposed in [43] to enhance

---

<sup>3</sup>To promote diversity in the obtained solutions, Das and Dennis's systematic approach [59] is adopted to generate structured reference points.

---

**Algorithm 1** The framework of MaOEA-RD

---

**Input:** Population size  $N$ , objectives  $m$ , the termination criterion;

**Output:** The optimized population  $P_t$ ;

- 1: Generate the initial population  $P$  and the reference vectors  $W$ ;
  - 2:  $HisCI = INF, HisW = W, Flag = 0, t = 0$ ;
  - 3: **while** the termination criterion is not met **do**
  - 4:   Crossover and mutation to generate offspring  $Q_t$ ;
  - 5:    $R_t = P_t \cup Q_t$ ;
  - 6:    $P_{t+1} = EnvironmentalSelection(W, R_t)$  (Algorithm 3);
  - 7:    $(W, HisCI, HisW, Flag) = VectorAdjustmentMechanism(P_{t+1}, W, HisCI, HisW, Flag)$  (Algorithm 4);
  - 8:    $t = t + 1$ ;
  - 9: **end while**
  - 10: Return  $P_t$ ;
- 

the selection pressure. The Gram-Schmidt orthogonalization procedure [61] is applied to get the  $CI$  values, which is shown in Algorithm 2.

In Algorithm 2, the convergence direction vector  $C$  is the same as ISNPS as an  $m$ -dimensional vector  $C = (1, 1, \dots, 1)$ . To make the first dimension of the rotated coordinates system coincide with  $C$ , a matrix  $A$  (line 1) is initialized and orthogonalized (lines 2-3). The resultant matrix  $P$  is used to transform the solutions in the objective space (line 4). After the matrix transformation, the first dimension of  $FP'$  is the  $CI$ , that is, the distance from the individual to the hyperplane<sup>4</sup>.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{Nm} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \cdots & 1 \end{pmatrix}. \quad (7)$$

---

<sup>4</sup>The hyperplane is perpendicular to the convergence direction vector and passes through the origin of coordinates.

---

**Algorithm 2** The calculation procedure of the rotated coordinates

---

**Input:** The convergence direction vector  $C$ , the coordinates of individuals  $FP$ ;

**Output:** The rotated coordinates of individuals  $FP'$ ;

- 1: Set a coordinates system matrix  $A$ , which is shown as formula 7;
  - 2: Make matrix  $A$  be an orthogonal matrix  $B=\{b_1, b_2, \dots, b_n\}$ , where  $b_i$  is produced by recursion formula 8;
  - 3: Make matrix  $B$  be a unit matrix and get the rotated coordinates matrix  $P$  by formula 9;
  - 4:  $FP' = P^T \cdot FP$ ;
  - 5: Return  $FP'$ .
- 

$$\begin{cases} b_1 = a_1, \\ b_2 = a_2 - \frac{[b_1, a_2]}{[b_1, b_1]} b_1, \\ \vdots \\ b_N = a_N - \frac{[b_1, a_N]}{[b_1, b_1]} b_1 - \frac{[b_2, a_N]}{[b_2, b_2]} b_2 - \dots - \frac{[b_{N-1}, a_N]}{[b_{N-1}, b_{N-1}]} b_{N-1}. \end{cases} \quad (8)$$

$$P = (p_1, p_2, \dots, p_N) = \left( \frac{b_1}{\|b_1\|}, \frac{b_2}{\|b_2\|}, \dots, \frac{b_N}{\|b_N\|} \right). \quad (9)$$

### 3.3. Environmental selection

The individuals can be distinguished by the values of  $CI$  after the rotated coordinates. However, the  $CI$  may not be accurate due to the variety of the PFs of the MOPs, which could lead to a misleading estimation of the individuals' convergence.

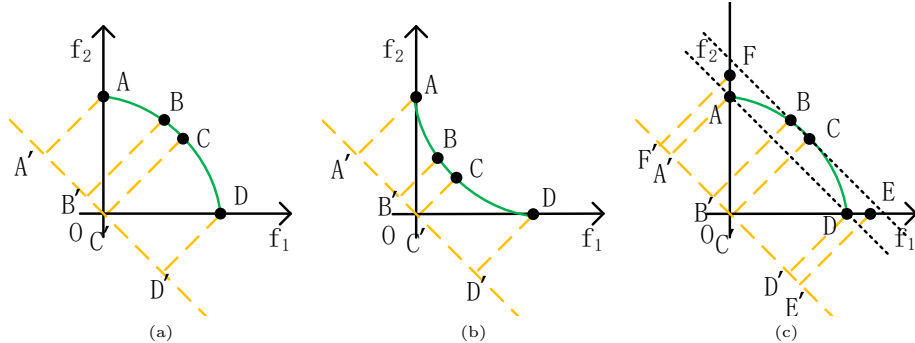


Figure 8: An example of an inaccurate estimation of the individuals' convergence.

In Figure 8, (a) and (b) are concave and convex problems, respectively. If we select individuals according to the distance from individual to hyperplane, individuals at both ends (A, D) or in the middle (B, C) will be favoured. Even worse, if there are dominance resistance solutions<sup>5</sup> in the population like (c), then (F, E) are chosen, which will prevent the population from reaching the true PF. Thus, a diversity maintenance mechanism to balance the effects of preferences arising from this distance is necessary. In our algorithm, the decomposition method is adopted. By using uniformly distributed reference vectors and a novel individual selection mechanism, the shortcoming can be addressed. Algorithm 3 is the environmental selection of MaOEA-RD.

In Algorithm 3, the Non-Dominated sorting (NDsort) [62] is carried out on the merged population  $R_t$ , and all the individuals before the critical layer are directly added to the  $P_{t+1}$  (line 3). Algorithm 2 is called to get the  $CI$  values. After associating individual<sup>6</sup> from  $F_c$  with the reference vectors  $W$  (line 4), all individuals in the critical layer<sup>7</sup> have the chance to be selected by our individuals' selection mechanism, which combines the uniformly distributed reference vectors to balance the influence of individuals' preferences caused by the  $CI$  values. For the individuals in the critical layer, we select the individual  $X$  with the smallest  $CI$  every time, and the individual, which is associated with the vector where  $X$  belongs, is ignored (line 6). If the size of  $P_{t+1}$  is smaller than  $N$ ,  $N - |P_{t+1}|$  individuals with the minimum  $PBI_m$  values are selected to fill  $P_{t+1}$  (line 9),  $|P_{t+1}|$  is the size of  $P_{t+1}$ . The  $PBI_m$  value is computed as formula 10, where  $d_1$  in the formula is replaced by the  $CI$ . Figure 9 is the illustration of the  $PBI_m$  value. The reason we take  $d_2$  into account is to balance the effects of preferences brought by  $d_1$ . Conversely, if the size of  $P_{t+1}$  is more than  $N$ , MaOEA-RD calculates the angle between all individuals in the population first, and then finds the individuals with the smallest angle and deletes the individual with the largest  $PBI_m$  value (line

---

<sup>5</sup>The dominance resistance solution is a nondominated solution, but one of its dimension objective values is the smallest in the population and the objective values of the remaining dimensions are very poor.

<sup>6</sup>The solution  $F(X_i)$  is associated with a vector  $W_j$  if  $F(X_i)$  has the smallest vertical distance from the vector  $W_j$ . If  $F(X_i)$  has multiple equal distances to multiple  $W_i$ ,  $F(X_i)$  is associated with the first of these vectors.

<sup>7</sup>Dividing  $2N$  solutions into set  $F_1, F_2, \dots, F_l$  and keeping  $N$  solutions. We call  $F_c$  the critical layer; the size of solutions from  $F_1$  to  $F_{c-1}$  is smaller than  $N$ , but the size of solutions from  $F_1$  to  $F_c$  is equal or more than  $N$ .

12). This process stops when the size of  $P_{t+1}$  is equal to  $N$ .

---

**Algorithm 3** The environmental selection in the proposed MaOEA-RD

---

**Input:** The reference vectors  $W$ , the merged population  $R_t$ ;

**Output:** The next population  $P_{t+1}$ ;

- 1:  $\{F_1, F_2, \dots, F_c, \dots\} = NDsort(R_t)$  [62];
  - 2: Get the rotated coordinates of individuals of  $F_c$  by Algorithm 2;
  - 3: The individuals before  $F_c$  are added into  $P_{t+1}$ ;
  - 4: Associate individuals of  $F_c$  with the reference vectors  $W$ ;
  - 5: **for** each  $W_i$  in  $W$  **do**
  - 6:   Add the individual from  $F_c$  that has the shortest distance into  $P_{t+1}$   
    and ignore other individuals associated with the same vector;
  - 7: **end for**
  - 8: **if**  $|P_{t+1}| < \text{the size of population } N$  **then**
  - 9:   Select  $N - |P_{t+1}|$  individuals from all the ignored individuals with the  
    minimum  $PBI_m$  values into  $P_{t+1}$ ;
  - 10: **end if**
  - 11: **if**  $|P_{t+1}| > \text{the size of population } N$  **then**
  - 12:   Delete individuals with the largest  $PBI_m$  values from the crowded area  
    until the size of  $P_{t+1}$  equals  $N$ ;
  - 13: **end if**
  - 14: Return  $P_{t+1}$ .
- 

$$PBI_m = d_1 + \theta d_2. \quad (10)$$

To facilitate our understanding of the environmental selection mechanism, a simple example is illustrated in Figure 10. We select the individual from the critical layer with the shortest distance to the hyperplane to ensure convergence, and all individuals associated with the reference vector, which is associated with the selected individual, are not considered in the next selection (Algorithm 3 line 6). For example, in Figure 10, A, B and C are associated with  $W_3$ , and A has the shortest distance. A is selected but B and C are ignored in the next selection. Assuming the population size is 3, under our environmental selection mechanism, A, D and H will be selected. Conversely, if we only consider the individuals with the smallest  $CI$  value, A, D and E will be selected, which seriously affects the distribution of the population. Figure 11 shows another case where the number of individuals is

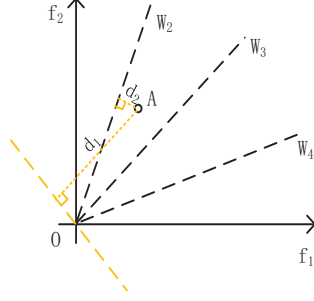


Figure 9: The illustration of  $d_1$  and  $d_2$ .

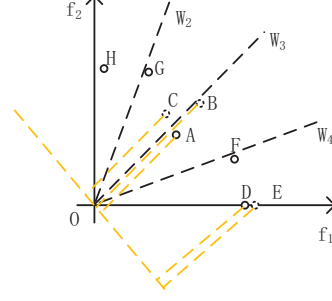


Figure 10: Illustration of the individual selection mechanism of case 1.

less than the population size after the first round of selection. In this case, we sort the values of  $PBI_m$  in ascending order, selecting the appropriate number of individuals to enter the next generation. Supposing the population size is 8 in Figure 11, J, B, C, G and H are selected for the first round of selection, respectively. Then, the ignored individuals are sorted in ascending order of their  $PBI_m$  values. Therefore, K, A and E are selected (Algorithm 3 line 9). For comparison, Figure 12 is the environmental selection result by the  $PBI$  approach. In the first round of selection, C, H, J, B and F are selected, and then E, G and D are selected according to the  $PBI$  values. From these two figures, we can find that using the  $PBI_m$  as the second selection criterion can better preserve the boundary solutions than using the  $PBI$ .

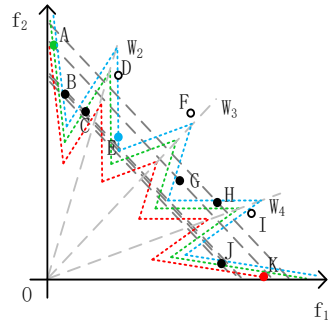


Figure 11: Illustration of the individual selection mechanism of case2.

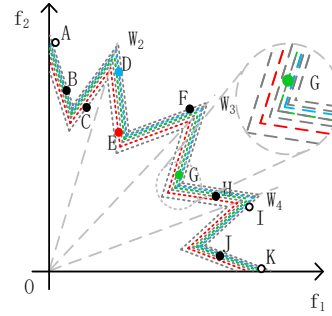


Figure 12: Illustration of the selection result with the  $PBI$  approach.

### 3.4. Reference vector adjustment mechanism

Decomposition-based MOEAs have shown their advantages in addressing MaOPs. However, when the optimization problem has an irregular PF (PFs, e.g., degenerate and disconnected ones), this decomposition decreases the algorithm's performance [44]. Many excellent adaptive decomposition-based algorithms have shown the effectiveness of adjusting reference vectors in solving MaOPs. However, there are two challenges that cannot be ignored in all algorithms with reference vector adjustment.

The first difficulty is the time to start to adjust the reference vectors. Some algorithms start adjusting vectors after a certain number of evolutions by setting a threshold value [52, 53]. In MaOEA-RD, it is expected that the algorithm will decide whether to adjust the reference vector according to the evolutionary state of the population. The second difficulty is the frequency of the reference vector adjustment. If the reference vectors are adjusted too frequently, the population does not have enough time to evolve. On the contrary, if the reference vectors are adjusted too slowly, due to the limited number of iterations, the desired effect may not be achieved. To overcome these difficulties, a reference vector adjustment mechanism is proposed in MaOEA-RD. Algorithm 4 is the details of the reference vector adjustment mechanism.

Algorithm 4 can be divided into two parts. The first part is to decide whether to adjust the reference vectors (lines 1-14), and the second part is how to adjust the reference vectors (lines 15-17).

Let's start with the first part of Algorithm 4. The reference vector adjustment strategy (Algorithm 5) may be carried out only after the population iterates  $n \times \varphi_1$  times (line 1), because it is not reasonable to adjust reference vectors in each iteration [46, 50, 51, 53, 57]. The  $\varphi_1$  is a parameter of MaOEA-RD, which is analysed in Section 4.4. We sum the  $CI$  values of all individuals as  $SumCI$  (line 2) and use the ratio of  $SumCI$  to  $HisCI$  to reflect the evolutionary potential of the population (line 3). The closer  $SumCI/HisCI$  is to 1, the smaller the convergence potential of the population. Therefore, when the ratio is around 1, the strategy of reference vector adjustment is activated and the current reference vectors are backed up to  $HisW$  (line 4). The purpose of backing up the reference vectors is to prevent the population from failing to evolve after the reference vector adjustment. When the ratio is not close to 1, the population has a lot of room for evolution, which means the reference vector adjustment strategy doesn't work. So the reference vector adjustment mechanism is deactivated (line 6). If the



last time reference vectors are adjusted hinders population evolution, the reference vectors are restored to  $HisW$  (lines 7-9).

---

**Algorithm 4** The reference vector adjustment mechanism in the proposed MaOEA-RD

---

**Input:** The population  $P_{t+1}$ , the reference vectors  $W$ , the  $HisCI$ , the  $HisW$ , the  $Flag$ ;

**Output:** The reference vectors  $W$ , the  $HisCI$ , the  $HisW$ , the  $Flag$ ;

```

1: if  $((t \bmod \varphi_1) == 0)$  then
2:   Sum the  $CI$  values of all individuals in the population as  $SumCI$ ;
3:   if  $((SumCI/HisCI) > \varphi_2$  and  $(SumCI/HisCI) < \varphi_3)$  then
4:      $Flag = 1, HisW = W$ ;
5:   else
6:      $Flag = 0$ ;
7:     if  $HisCI < SumCI$  then
8:        $W = HisW$ ;
9:     end if
10:  end if
11:  if  $SumCI < HisCI$  then
12:     $HisCI = SumCI$ ;
13:  end if
14: end if
15: if  $(Flag)$  then
16:    $W = Adjustment(P_{t+1}, W)$  (Algorithm 5);
17: end if
18: Return  $W, HisCI, HisW, Flag$ ;
```

---

Next, let's focus on Algorithm 5.  $NDsort$  [62] is used to get all the nondominated solutions first (line 1). Then, all the nondominated solutions are associated with vectors (line 2), and the vectors without solutions are removed (line 3). Next, the same number of vectors are generated as deleted (lines 4-12). The new reference vectors should be generated between the vectors which are useful and in a relatively sparse area. Therefore, the nearest vector<sup>8</sup> for each of the reference vectors is found (line 6), and then we find the maximum from those, which must be a pair of vectors (line 8). The new

---

<sup>8</sup>The Angle between vectors is adopted.

vector is the line between the midpoint of these two vectors and the origin (line 9). This process stops when the number of reference vectors is equal to the size of  $P_{t+1}$ .

---

**Algorithm 5** The reference vector adjustment strategy in the proposed MaOEA-RD

---

**Input:** The population  $P_{t+1}$ , the reference vectors  $W$ ;

**Output:** The adjustment reference vectors  $W$ ;

- 1:  $\{F_1, F_2, \dots, F_c, \dots\} = NDsort(P_t + 1)$  [62];
  - 2: Associate individuals of  $F_1$  with the reference vectors  $W$ ;
  - 3: Delete all the reference vectors that are not associated with individuals.
  - 4: **while**  $|W| < |P_{t+1}|$  **do**
  - 5:   **for** each  $W_i$  in  $W$  **do**
  - 6:     Find the closest reference vector ( $W_{ci}$ ) to  $W_i$  and record the angle between them;
  - 7:   **end for**
  - 8:   Find the largest angle from the angle recorded by each reference vector and get the corresponding set of vectors ( $W_{cj}, W_j$ );
  - 9:    $W_{new} = (W_{cj} + W_j)/2$ ;
  - 10:    $W = W \cup W_{new}$ ;
  - 11: **end while**
  - 12: Return  $W$ ;
- 

Figure 13 is an example of the vectors' adjustment. Firstly,  $NDsort$  is carried out to find effective reference vectors (Figure 13 (a)). The vector that contains no associated nondominated solutions is very likely to contain no Pareto optimal solutions; therefore,  $W_3$  is deleted (Figure 13 (b)). Finally, the new reference vector is generated in the most sparse area where effective vectors exist. (Figure 13 (c), in this example, the  $W_{new}$  also can be generated between  $f_2$  and  $W_2$  because they are in the most sparse area where there are effective vectors.)

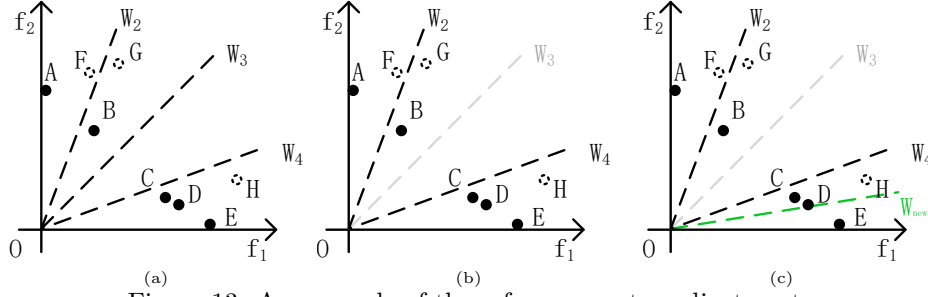


Figure 13: An example of the reference vector adjustment.

#### 4. Experimental design and analysis

To validate and analyze the performance of MaOEA-RD, the DTLZ and WFG benchmark instances, which are widely used to detect the performance of algorithms in many-objective optimization, are employed. Characteristics of all instances are summarized in Table 1. The number of objectives is set as 5, 8, 10 and 15, respectively. For all the DTLZ problems, the length of decision variables is set to  $k + m - 1$ , where  $m$  denotes the number of objectives; and  $k$  is set to 5 for DTLZ1, 20 for DTLZ7, and 10 for the others. Regarding the WFG problems, the length of decision variables is set to  $k + l$ , where  $k$  and  $l$  are set to  $m - 1$  and 10, respectively. Nine state-of-the-art MaOEAs are selected for comparative experiments; they are ISNPS [43], RVEA [52], NSGA-III [16], VaEA [63], MOEA/DD [17], RVEA\*<sup>9</sup> [52], ANSGA-III [50], MOEA/D-AWA [51] and MOEA/D-URAW [55]. All algorithms are run in the same experimental platform PlatEMO [64] in MATLAB R2016a except ISNPS, which is implemented in the OTL [65].

##### 4.1. Performance metric

The *Hypervolume* (HV) [29] metric is adopted to compare the performance of different algorithms. HV can measure both convergence and diversity of a solution set, which has been accepted by peers and is used as a common measure of an algorithm's performance.

HV measures the volume of the hypercube dominated by an approximation set [67]. It can be expressed as formula 11, where  $S(X_i)$  is the hypercube

<sup>9</sup>RVEA\* is the version of RVEA with reference vector adaptive adjustment.

Table 1: Features of DTLZ and WFG problems [66]

Features	Problems
Multi-modal	DTLZ1, DTLZ3, DTLZ6
Biased	DTLZ4, WFG1, WFG7, WFG8, WFG9
Non-separable	WFG2, WFG3, WFG6, WFG8, WFG9
Deceptive	WFG5, WFG9
Degenerate	DTLZ5, DTLZ6, DTLZ7, WFG3
Disconnect	DTLZ6, WFG2
Linear	DTLZ1, WFG3
Convex	WFG2
Mixed	DTLZ7, WFG1
Concave	DTLZ2, DTLZ3, DTLZ4, WFG5, WFG6, WFG7, WFG8, WFG9

bounded by a solution  $X_i$  in the PF furnished by the algorithm and a reference point  $r$ . The larger the HV value, the better the result.

$$HV = \cup_{X_i \in PF} S(X_i). \quad (11)$$

The reference point  $r$  settings used to calculate the HV indicator are shown in Table 2.

Table 2: The reference point  $r$  for calculating the HV indicator

Objectives	DTLZ1-4
5	(0.5500, 0.5500, 0.5500, 0.5500, 0.5500)
8	(0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500)
10	(0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500)
15	(0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500, 0.5500)

Objectives	DTLZ5 and DTLZ6
5	(0.3889,0.3889, 0.5500,0.7778,1.1000)
8	(0.1375,0.1375,0.1945,0.2750, 0.3889,0.5500,0.7778,1.1000)
10	(0.0687,0.0687,0.0972,0.1375,0.1945, 0.2750,0.3889,0.5500,0.7778,1.1000)
15	(0.0122,0.0122,0.0172,0.0243,0.0344,0.0486,0.0687, 0.0972,0.1375,0.1945,0.2750,0.3889,0.5500,0.7778,1.1000)
Objectives	DTLZ7
5	(0.9453,0.9453, 0.9453,0.9453,11.0000)
8	(0.9453,0.9453,0.9453,0.9453, 0.9453,0.9453,0.9453,17.6000)
10	(0.9453,0.9453,0.9453,0.9453,0.9453, 0.9453,0.9453,0.9453,0.9453,22.0000)
15	(0.9453,0.9453,0.9453,0.9453,0.9453,0.9453,0.9453, 0.9453,0.9453,0.9453,0.9453,0.9453,0.9453,33.0000)
Objectives	WFG1
5	(2.1876,4.3752, 6.5642,8.7547,10.9999)
8	(2.1784,4.3567,6.5365,8.7178, 10.9005,13.0849,15.2709,17.5996)
10	(2.1723,4.3445,6.5182,8.6933,10.8699, 13.0481,15.2280,17.4094,19.5925,21.9990)
15	(2.1571,4.3141,6.4726,8.6325,10.7939,12.9568,15.1214, 17.2876,19.4554,21.6248,23.7959,25.9686,28.1430,30.3191,32.9951)
Objectives	WFG3
5	(0.2750,0.5500, 1.6500,4.4000,11.0000)
8	(0.0344, 0.0688,0.2063,0.5500, 1.3750,3.3000,7.7000,17.6000)
10	(0.0086,0.0172,0.0516,0.1375,0.3438, 0.8250,1.9250,4.4000,9.9000,22.0000)
15	(0.0003,0.0005,0.0016,0.0043,0.0107,0.0258,0.0602, 0.1375,0.3094,0.6875,1.5125,3.3000,7.1500,15.4000,33.0000)

Objectives	WFG2 and WFG4-9
5	(2.2000,4.4000, 6.6000,8.8000,11.0000)
8	(2.2000,4.4000,6.6000,8.8000, 11.0000,13.2000,15.4000,17.6000)
10	(2.2000,4.4000,6.6000,8.8000,11.0000, 13.2000,15.4000,17.6000,19.8000,22.0000)
15	(2.2000,4.4000,6.6000,8.8000,11.0000,13.2000,15.4000, 17.6000,19.8000,22.0000,24.2000,26.4000,28.6000,30.8000,33.0000)

#### 4.2. General parameters

- 1) Population size: The population size is set to be the same as the number of the reference vectors. In particular,  $N$  is set to be 210, 156, 275 and 135 for  $m = \{5, 8, 10, 15\}$ , respectively.
- 2) Reproduction operators: For the *SBX*, the crossover probability  $p_c$  is set to 1.0 and the distribution index  $\eta_c$  is set to 30. As for the polynomial mutation, the probability  $p_m$  and distribution index  $\eta_m$  are set to be  $1/n$  ( $n$  represents decision number) and 20, respectively.
- 3) Number of evaluation: Different evaluation times are adopted for different instances, and the specific values are shown in Table 3. All algorithms are run 30 times independently.
- 4) Specific parameter settings in each algorithm: The parameters in the algorithm are set according to the recommendations in the original publications. The  $\alpha$  of ISNPS is shown in Table 4 [43]. For MOEA/DD, the  $T$ , the  $\theta$  and the  $\delta$  are set to 20, 5 and 0.9, respectively [17]. For RVEA and RVEA\*, the  $\alpha$  is set to 2 and the  $f_r$  is set to 0.1 [52]. For MOEA/D-AWA, the rate update weight is set to 0.05; the maximal number of subproblem adjusted  $nus$  is set to rate update weight  $\times N$ , the rate  $evol$  is set to 0.8 [51]. For MOEA/D-URAW, the  $\delta$  and the  $nr$  are set to 0.9 and 2, respectively [55].

Table 3: Evaluation times

Objectives	DTLZ1	DTLZ2	DTLZ3	DTLZ4-7	WFG1-9
5	1.2600e+5	7.3500e+4	2.1000e+5	2.1000e+5	1.5750e+5
8	1.1700e+5	7.8000e+4	1.5600e+5	1.9500e+5	2.3400e+5
10	2.7500e+5	2.0625e+5	4.1250e+5	5.5000e+5	5.5000e+5
15	2.2050e+5	1.3500e+5	2.7000e+5	4.0500e+5	4.0500e+5

Table 4: The  $\alpha$  for ISNPS

Objectives	DTLZ1	DTLZ2-4	DTLZ5-6	DTLZ7	WFG1	WFG2	WFG3	WFG4-9
5	22.5	21.9	0.81	9.5	10.1	12.5	0.95	11.2
8	35.6	37	1.5	39	16.7	23.5	1.56	22.9
10	38	39	1.7	42	19.3	27.1	1.91	29.7
15	41	41.5	1.9	45	23	30.4	2.31	31.4

#### 4.3. Comparisons with state-of-the-art MaOEAs

In this section, MaOEA-RD is compared with nine MaOEAs. They are ISNPS [43], RVEA [52], NSGA-III [16], VaEA [63], MOEA/DD [17], RVEA\* [52], ANSGA-III [50], MOEA/D-AWA [51] and MOEA/D-URAW [55]. Among these algorithms, RVEA\*, ANSGA-III, MOEA/D-AWA and MOEA/D-URAW are the state-of-the-art decomposition-based MaOEAs with reference vector adjustment. To have statistically comprehensive conclusions, the Wilcoxon's Rank test [68] at a 0.05 significance level is adopted to test the significant difference between the data obtained by paired algorithms. Table 5 and Table 6 are the results of the HV indicator<sup>10</sup>. The best performing algorithm is highlighted in dark color. "-" and "+" represent significant differences where, compared by their means, MaOEA-RD is better or worse than other algorithms, while "=" indicates that no significant difference is detected.

As can be seen from Table 5, MaOEA-RD performed the best on DTLZ1 and DTLZ6, and performed the second best on DTLZ5 and some objectives of DTLZ7. The reasons can be summarized as follows: First, the true PF of DTLZ1 is exactly parallel to the hyperplane we constructed, so our selection of individuals according to the  $CI$  values is completely unaffected by the

<sup>10</sup>We recommend normalizing the population on DTLZ7 and WFG problems whose objective values are disparately scaled. The reason can be found in [69].

deficiencies mentioned in subsection 3.3. Secondly, for problems with irregular PFs, the algorithm based on adjusting the reference vectors has more advantages than the algorithm based on decomposition. This is why RVEA, NSGA-III, VaEA and MOEA/DD were not good on DTLZ5, DTLZ6 and DTLZ7 problems but MOEA/D-AWA, MOEA/D-URAW were competitive on some DTLZ6 and DTLZ7 problems.

As can be seen from Table 6, different algorithms have their own strengths on different problems. From the statistics, MaOEA-RD performed better on HV scores in WFG4-WFG8 out of a few problems. A very strange phenomenon is that our algorithm does not perform well on WFG1-WFG3 problems, whose PFs are irregular. The reasons for this are explained in the experiment in subsection 4.4.

By comparing the results of RVEA and RVEA\*, NSGA-III and ANSGA-III, MOEA/DD and MOEA/D-URAW, we find that the decomposition-based MaOEAs with the reference vector adjustment are effective when solving problems with irregular PFs, but compromise the performance on regular PFs. In contrast, MaOEA-RD has better robustness and has good performance for both regular and irregular PFs.

Table 5: Mean and standard deviation of HV values on DTLZ1-DTLZ7 problems

Instances	M	MaOEA-RD	ISNPS [43]	RVEA [52]	NSGA-III [16]	VaEA [63]	MOEA/DD [17]	RVEA* [52]	ANSGA-III [50]	MOEA/D-AWA [51]	MOEA/D-URAW [55]
DTLZ1	5	4.9316e-2	4.8908e-2	4.9310e-2	4.9309e-2	4.5443e-2	4.9312e-2	4.9247e-2	4.9265e-2	4.9301e-2	4.9110e-2
		(6.17e-6)	(2.29e-4)	(8.82e-6)	(9.64e-6)	(3.09e-3)	(7.12e-6)	(1.64e-5)	(7.37e-5)	(1.09e-5)	(4.24e-5)
	8	8.3532e-3	8.2713e-3	8.3532e-3	8.0976e-3	8.3509e-3	8.3526e-3	8.3503e-3	8.0976e-3	8.3247e-3	8.3046e-3
		(5.05e-7)	(3.39e-5)	(3.99e-7)	(6.51e-4)	(7.27e-6)	(5.42e-7)	(1.56e-6)	(6.51e-4)	(1.41e-5)	(1.60e-5)
	10	2.5322e-3	2.5235e-3	2.5322e-3	2.5321e-3	2.4375e-3	2.5321e-3	2.5321e-3	2.5197e-3	2.5305e-3	2.5281e-3
		(2.77e-8)	(2.80e-6)	(3.46e-8)	(3.47e-8)	(4.12e-5)	(6.52e-8)	(6.18e-8)	(3.66e-5)	(1.30e-6)	(1.38e-6)
	15	1.2747e-4	1.2607e-4	1.2747e-4	1.2616e-4	1.2488e-4	1.2745e-4	1.2745e-4	1.2497e-4	1.2532e-4	1.2674e-4
		(2.10e-9)	(3.08e-7)	(2.43e-9)	(3.11e-6)	(9.58e-7)	(1.26e-8)	(6.29e-8)	(1.39e-6)	(1.65e-6)	(2.52e-7)
	5	1.3091e+0	1.2627e+0	1.3080e+0	1.3074e+0	1.2770e+0	1.3078e+0	1.3015e+0	1.3050e+0	1.3053e+0	1.2881e+0
		(4.88e-4)	(6.69e-3)	(4.10e-4)	(6.66e-4)	(4.69e-3)	(7.51e-4)	(1.80e-3)	(2.63e-3)	(1.73e-3)	(4.22e-3)
	8	1.9797e+0	1.9320e+0	1.9804e+0	1.8811e+0	1.9499e+0	1.9805e+0	1.9677e+0	1.8811e+0	1.9341e+0	1.9098e+0
		(6.16e-4)	(2.20e-2)	(6.30e-4)	(1.24e-1)	(7.88e-2)	(6.83e-4)	(2.30e-3)	(1.24e-1)	(7.55e-2)	(1.58e-2)
DTLZ2	10	2.5150e+0	2.4940e+0	2.5153e+0	2.3859e+0	2.4841e+0	2.5153e+0	2.5077e+0	2.3859e+0	2.4974e+0	2.4380e+0
		(3.38e-4)	(8.49e-3)	(4.24e-4)	(1.08e-1)	(6.34e-2)	(3.17e-4)	(9.11e-4)	(1.08e-1)	(3.92e-2)	(1.28e-2)
	15	4.1377e+0	3.8620e+0	4.1356e+0	3.6591e+0	3.9807e+0	4.1382e+0	4.1321e+0	3.6591e+0	3.7883e+0	3.7420e+0
		(5.46e-4)	(5.74e-2)	(5.62e-3)	(1.17e-1)	(1.05e-1)	(2.98e-4)	(1.72e-3)	(1.17e-1)	(2.64e-1)	(9.97e-2)



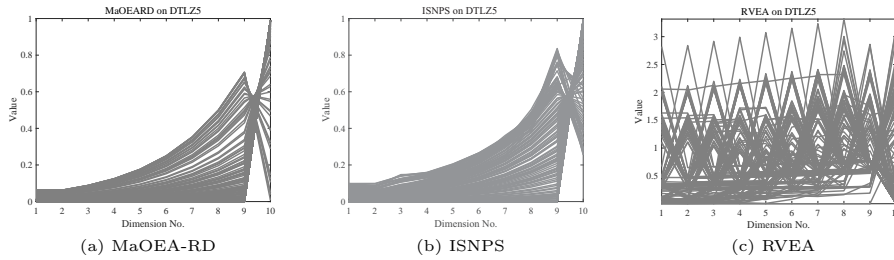
Instances	M	MaOEA-RD	ISNPS [43]	RVEA [52]	NSGA-III [16]	VaEA [63]	MOEA/DD [17]	RVEA* [52]	ANSGA-III [50]	MOEA/D-AWA [51]	MOEA/D-URAW [55]
DTLZ3	5	1.3061e+0	1.2997e+0	1.3075e+0	1.3046e+0	1.0448e+0	1.3072e+0	1.2993e+0	1.2836e+0	1.3025e+0	1.2642e+0
		(1.66e-3)	(5.35e-3)	(1.82e-3) +	(3.27e-3) =	(1.26e-1) -	(1.46e-3) =	(4.00e-3) -	(2.17e-2) -	(1.89e-3) -	(4.66e-3) -
	8	1.9738e+0	1.9364e+0	1.9731e+0	1.6281e+0	1.787e+0	1.9765e+0	1.8049e+0	1.6281e+0	1.9218e+0	1.8390e+0
		(6.29e-3)	(7.58e-3)	(5.33e-3) =	(2.50e-1) -	(4.26e-1) -	(3.42e-3) =	(5.01e-1) -	(2.50e-1) -	(4.11e-2) -	(3.95e-2) -
	10	2.5147e+0	2.4871e+0	2.5148e+0	2.4164e+0	4.3984e-1	2.5148e+0	2.4914e+0	2.1116e+0	2.4509e+0	2.3267e+0
		(9.59e-4)	(7.39e-3)	(7.63e-4) =	(1.44e-1) -	(6.23e-1) -	(4.55e-4) =	(8.97e-3) -	(6.05e-1) -	(8.52e-2) -	(6.69e-2) -
	15	4.1143e+0	3.7299e+0	4.1294e+0	0.0000e+0	3.2529e+0	4.1377e+0	4.1188e+0	0.0000e+0	3.3251e+0	3.4605e+0
		(8.75e-2)	(1.09e-1)	(3.00e-2) =	(0.00e+0) -	(9.58e-1) -	(6.07e-4) =	(7.56e-3) +	(0.00e+0) -	(3.88e-1) -	(9.89e-2) -
	5	1.3085e+0	1.2659e+0	1.3087e+0	1.2929e+0	1.1808e+0	1.3091e+0	1.2965e+0	1.2946e+0	1.2215e+0	1.3051e+0
		(5.56e-4)	(6.52e-3)	(6.62e-4) =	(6.02e-2) =	(1.81e-2) -	(7.29e-4) +	(3.73e-2) -	(5.28e-2) =	(7.35e-2) =	(2.38e-3) -
	8	1.9508e+0	1.9840e+0	1.9807e+0	1.8946e+0	1.9353e+0	1.9809e+0	1.9771e+0	1.8933e+0	1.9200e+0	1.9396e+0
		(4.73e-2)	(2.42e-2)	(6.43e-4) +	(1.02e-1) =	(8.73e-3) =	(4.06e-4) +	(2.42e-3) =	(1.54e-1) =	(7.61e-2) =	(2.33e-2) =
	10	2.5151e+0	2.4951e+0	2.5152e+0	2.4737e+0	2.4430e+0	2.5154e+0	2.5121e+0	2.4293e+0	2.4855e+0	2.4857e+0
		(2.44e-4)	(3.05e-3)	(6.05e-4) =	(7.24e-2) -	(9.59e-3) -	(4.88e-4) =	(7.92e-4) -	(1.28e-1) -	(3.98e-2) -	(9.17e-3) -
	15	4.1306e+0	3.7829e+0	4.1309e+0	4.0427e+0	4.0238e+0	4.1347e+0	4.1373e+0	3.9988e+0	4.0588e+0	4.0247e+0
		(1.05e-2)	(4.13e-2)	(8.98e-3) =	(7.83e-2) -	(8.00e-3) -	(7.49e-3) =	(3.94e-3) +	(1.24e-1) -	(6.79e-2) -	(2.44e-2) -
DTLZ5	5	8.9138e-3	9.7747e-3	7.7509e-3	3.8026e-3	7.1080e-3	8.2068e-3	7.7337e-3	3.8026e-3	7.5093e-3	8.6767e-3
		(1.55e-4)	(8.25e-5) +	(1.43e-3) -	(3.18e-3) -	(2.98e-4) -	(4.01e-5) -	(2.46e-4) -	(3.18e-3) -	(3.95e-4) -	(1.29e-4) =
	8	1.9172e-5	1.9933e-5	1.6821e-5	4.8102e-6	1.6824e-5	1.7782e-5	1.7546e-5	9.9422e-7	1.8479e-5	1.8419e-5
		(2.95e-7)	(6.40e-7) +	(7.28e-11) -	(5.79e-6) -	(3.31e-8) -	(8.13e-8) -	(3.95e-7) -	(2.11e-6) -	(9.97e-8) -	(1.82e-7) -
	10	6.1584e-8	6.5084e-8	5.6445e-8	2.7034e-12	5.6120e-8	5.8292e-8	5.6968e-8	1.4305e-10	5.9676e-8	5.8353e-8
		(5.21e-10)	(3.41e-10) +	(5.00e-10) -	(1.05e-11) -	(1.12e-10) -	(1.53e-10) -	(7.65e-10) -	(5.51e-10) -	(2.27e-10) -	(5.62e-9) -
	15	8.7480e-17	8.873e-17	8.4445e-17	8.4575e-17	8.4309e-17	8.5204e-17	8.4834e-17	5.6168e-17	3.9584e-17	8.5873e-17
		(6.36e-19)	(2.57e-18) =	(3.19e-19) -	(4.02e-19) -	(4.05e-20) -	(6.26e-19) -	(5.51e-19) -	(3.33e-17) -	(4.30e-17) -	(2.14e-18) -
	5	8.3943e-3	7.7534e-3	6.8488e-3	2.1798e-3	6.2186e-3	8.2106e-3	7.1948e-3	2.1798e-3	7.3174e-3	8.3694e-3
		(2.20e-4)	(1.38e-4)	(1.45e-3) -	(2.85e-3) -	(1.01e-3) -	(2.35e-5) -	(2.60e-4) -	(2.85e-3) -	(2.55e-4) -	(1.91e-4) =
	8	1.8696e-5	1.5190e-5	1.7421e-5	1.2336e-5	7.8419e-6	1.7738e-5	1.6970e-5	3.6744e-6	1.8455e-5	1.7544e-5
		(5.39e-7)	(1.20e-6)	(2.36e-7) -	(7.70e-6) -	(8.68e-6) -	(6.39e-8) -	(2.44e-7) -	(6.91e-6) -	(7.45e-8) -	(2.55e-6) -
	10	6.0100e-8	4.7640e-8	5.7031e-8	2.0359e-9	0.0000e+0	5.8308e-8	5.6294e-8	0.0000e+0	5.9730e-8	5.3314e-8
		(1.19e-9)	(2.41e-9)	(5.73e-10) -	(7.88e-9) -	(0.00e+0) -	(1.48e-10) -	(2.54e-10) -	(0.00e+0) -	(1.46e-10) =	(7.89e-9) -
	15	8.6570e-17	6.4301e-17	8.4307e-17	7.8693e-17	8.4428e-17	8.5382e-17	8.4322e-17	1.1229e-17	7.4475e-18	7.1212e-17
		(1.06e-18)	(4.19e-18)	(2.59e-19) -	(2.18e-17) -	(2.38e-19) -	(2.70e-19) -	(3.11e-20) -	(2.96e-17) -	(2.24e-17) -	(2.49e-17) -
DTLZ7	5	2.2401e+0	1.5594e+0	2.2239e+0	2.1592e+0	2.1826e+0	9.2611e-1	2.4071e+0	2.1592e+0	1.7590e+0	2.3487e+0
		(1.37e-1)	(8.15e-2)	(4.13e-2) -	(5.91e-2) -	(2.20e-2) -	(2.89e-1) -	(2.31e-2) +	(5.91e-2) -	(9.72e-2) -	(1.70e-2) +
	8	2.2911e+0	5.2524e-1	1.9527e+0	2.3950e+0	2.0076e+0	2.2395e-1	2.2110e+0	2.0837e+0	1.1852e+0	2.2179e+0
		(7.47e-2)	(3.49e-1)	(1.95e-1) -	(4.47e-2) +	(4.40e-2) -	(3.17e-1) -	(1.62e-1) =	(7.92e-2) -	(4.63e-1) -	(8.47e-2) -
	10	2.3647e+0	1.8657e-1	1.9880e+0	2.3327e+0	1.8841e+0	9.8472e-4	2.1053e+0	2.4233e+0	1.6292e+0	2.5472e+0
		(5.82e-2)	(1.04e-1)	(3.01e-1) -	(9.23e-2) =	(9.69e-2) -	(6.02e-4) -	(2.15e-1) -	(8.47e-2) +	(3.57e-1) -	(5.42e-2) +
	15	2.3194e+0	4.8024e-1	1.7994e+0	1.9108e+0	1.5430e+0	5.6746e-6	1.2055e+0	1.3737e+0	1.6012e+0	2.3820e+0
		(6.94e-2)	(1.75e-1)	(7.89e-2) -	(2.20e-1) -	(4.71e-2) -	(1.16e-6) -	(2.02e-1) -	(5.80e-1) -	(4.16e-1) -	(7.35e-2) +
+/=-	-	-/-/-	3/24/1	3/14/11	1/22/5	0/26/2	5/15/8	3/23/2	1/25/2	0/25/3	3/24/1

Table 6: Mean and standard deviation of HV values on WFG1-WFG9 problems

Instances	M	MaOEA-RD	ISPNs [43]	RVEA [52]	NSGA-III [16]	VaEA [63]	MOEA/DD [17]	RVEA* [52]	ANSGA-III [50]	MOEA/D-AWA [51]	MOEA/D-URAW [55]
WFG1	5	6.0423e+3 (5.54e-1)	1.1564e+3 (1.60e-2)-	6.0378e+3 (1.63e+0) -	6.0120e+3 (6.77e+1) -	5.4248e+3 (1.64e+2) -	5.8158e+3 (1.13e+2) -	5.3408e+3 (1.83e+2) -	5.7714e+3 (1.58e+2) -	5.9424e+3 (1.61e+2) -	5.9868e+3 (3.38e+0) -
	8	2.0258e+7 (9.03e+5)	4.4100e+6 (2.03e+5)-	2.0511e+7 (6.59e+5) =	2.0729e+7 (1.00e+3) +	2.0721e+7 (4.52e+3) +	2.0500e+7 (5.83e+4) +	2.0152e+7 (6.24e+5) -	2.0717e+7 (6.81e+3) +	2.0456e+7 (5.24e+2) +	2.0450e+7 (2.36e+3) +
	10	8.5436e+9 (2.81e+8)	2.0133e+8 (6.37e+7)-	8.6521e+9 (2.40e+6) =	8.6619e+9 (1.52e+6) +	7.9611e+9 (3.80e+8) -	8.5766e+9 (1.57e+8) +	8.4435e+9 (1.83e+8) -	8.5200e+9 (9.64e+5) +	8.5200e+9 (1.62e+4) -	8.5188e+9 (5.22e+5) -
	15	1.3904e+17 (7.85e+13)	2.8431e+16 (3.35e+15)-	1.3904e+17 (3.66e+13) =	1.3913e+17 (3.11e+13) +	1.3914e+17 (1.04e+12) +	1.3726e+17 (4.60e+15) -	1.3507e+17 (9.99e+13) -	1.3911e+17 (3.40e+13) +	1.3424e+17 (4.83e+15) -	1.3547e+17 (7.56e+12) -
WFG2	5	6.1623e+3 (4.13e+0)	5.6622e+3 (6.01e+1)-	6.1482e+3 (6.00e+0) -	6.1467e+3 (8.08e+0) -	6.1616e+3 (3.75e+0) =	6.0403e+3 (2.90e+1) -	5.9747e+3 (8.57e+0) -	6.1466e+3 (8.37e+0) -	6.0060e+3 (8.98e+0) -	6.0186e+3 (1.99e+0) -
	8	2.1122e+7 (3.88e+5)	1.9254e+7 (1.68e+6)-	2.1862e+7 (9.78e+4) +	2.2075e+7 (4.67e+4) +	2.2055e+7 (1.87e+4) +	2.1257e+7 (1.18e+5) =	2.0492e+7 (2.75e+4) -	2.1991e+7 (7.36e+4) +	2.0577e+7 (2.41e+4) -	2.0615e+7 (8.42e+3) -
	10	9.1141e+9 (4.76e+8)	9.3579e+9 (5.86e+7)+	9.5675e+9 (2.23e+7) +	9.6181e+9 (1.05e+7) +	9.5951e+9 (1.49e+7) +	9.2551e+9 (4.59e+7) =	8.5270e+9 (1.80e+7) -	9.6222e+9 (1.09e+7) +	8.6047e+9 (5.18e+6) -	8.6075e+9 (5.73e+6) -
	15	1.4650e+17 (2.43e+16)	1.6759e+17 (8.05e+15)+	1.7608e+17 (7.38e+14) +	1.7791e+17 (6.44e+14) +	1.7854e+17 (1.44e+14) +	1.7110e+17 (1.72e+15)+	1.3684e+17 (2.25e+14) =	1.7787e+17 (1.18e+15) +	1.3749e+17 (1.20e+14) =	1.3757e+17 (1.43e+14) =
WFG3	5	2.9256e+0 (2.02e-1)	2.3247e-1 (9.42e-2)-	2.0124e+0 (1.57e-1) -	2.0503e+0 (1.40e-1) -	1.5283e+0 (3.10e-1) -	1.5951e+0 (2.32e-1) -	1.6206e+0 (2.81e-1) -	1.9041e+0 (2.14e-1) -	2.5246e+0 (2.26e-1) -	2.8381e+0 (9.97e-2) =
	8	6.3310e-3 (5.72e-3)	6.0605e-5 (1.81e-4)-	0.0000e+0 (0.00e+0) -	1.4476e-2 (3.86e-3) +	1.3483e-2 (3.87e-3) +	3.0891e-3 (4.13e-3) =	5.6117e-4 (1.03e-3) -	8.9334e-3 (5.57e-3) =	2.1101e-2 (3.34e-3) +	1.2698e-2 (5.55e-3) +
	10	7.0402e-6 (1.52e-5)	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0) -	6.1041e-6 (9.73e-6) =	3.2482e-5 (1.04e-5) +	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	4.3401e-6 (7.95e-6) =	4.6734e-5 (2.40e-6) +	6.7374e-6 (1.03e-5) =
	15	5.0764e-20 (1.97e-19)	0.0000e+0 (0.00e+0)=	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	3.0624e-17 (7.96e-17) =	0.0000e+0 (0.00e+0) -
WFG4	5	4.9931e+3 (6.05e+0)	4.3801e+3 (1.48e+2)-	5.0016e+3 (5.73e+0) +	4.9127e+3 (1.93e+1) -	4.7962e+3 (2.08e+1) -	4.8433e+3 (1.40e+1) -	4.8204e+3 (1.81e+1) -	4.9128e+3 (2.00e+1) -	4.9821e+3 (2.43e+1) =	4.9612e+3 (7.95e+0) -
	8	2.0414e+7 (2.10e+4)	1.7259e+7 (5.93e+5)-	2.0382e+7 (1.33e+4) -	2.0326e+7 (8.70e+3) -	1.9864e+7 (1.24e+5) -	1.7068e+7 (3.90e+5) -	1.9470e+7 (8.44e+4) -	1.9097e+7 (4.35e+5) -	2.0201e+7 (2.88e+4) -	2.0154e+7 (2.97e+4) -
	10	9.3372e+9 (8.05e+6)	8.2844e+9 (1.63e+8)-	9.3318e+9 (2.49e+6) -	9.2725e+9 (1.08e+8) -	8.9543e+9 (3.76e+7) -	7.6965e+9 (1.71e+8) -	8.8301e+9 (4.60e+7) -	8.8196e+9 (1.33e+8) -	9.2101e+9 (2.27e+7) -	9.1854e+9 (1.38e+7) -
	15	1.7732e+17 (2.03e+13)	1.4380e+17 (7.74e+15)-	1.7727e+17 (3.77e+13) -	1.7384e+17 (3.47e+15) -	1.7026e+17 (5.04e+14) -	1.0282e+17 (7.76e+15) -	1.7269e+17 (6.59e+14) -	1.5731e+17 (5.51e+15) -	1.7234e+17 (2.19e+14) -	1.7149e+17 (5.00e+14) -
WFG5	5	4.7085e+3 (1.80e+0)	4.2318e+3 (5.85e+1)-	4.7070e+3 (2.61e+0) =	4.7085e+3 (3.15e+0) =	4.5733e+3 (2.37e+1) -	4.5595e+3 (1.36e+1) -	4.5791e+3 (2.05e+1) -	4.6602e+3 (1.63e+1) -	4.6743e+3 (2.03e+1) -	4.6205e+3 (1.35e+1) -
	8	1.9110e+7 (4.64e+3)	1.7286e+7 (3.06e+5)-	1.9095e+7 (6.75e+3) -	1.8998e+7 (5.36e+3) -	1.8603e+7 (8.26e+4) -	1.7342e+7 (3.49e+5) -	1.8469e+7 (5.57e+4) -	1.8338e+7 (1.04e+5) -	1.8759e+7 (6.14e+4) -	1.8658e+7 (8.79e+4) -
	10	8.7216e+9 (1.48e+6)	7.9258e+9 (2.04e+8)-	8.7179e+9 (2.19e+6) -	8.7149e+9 (1.41e+6) -	8.4231e+9 (3.04e+7) -	7.4029e+9 (9.43e+7) -	8.4350e+9 (1.98e+7) -	8.4108e+9 (1.17e+8) -	8.4786e+9 (2.33e+7) -	8.4970e+9 (3.26e+7) -
	15	1.6424e+17 (1.68e+13)	1.4081e+17 (5.09e+15)-	1.6425e+17 (2.11e+13) =	1.6397e+17 (9.98e+14) -	1.5695e+17 (4.45e+14) -	1.1026e+17 (4.87e+15) -	1.6174e+17 (1.51e+14) -	1.6193e+17 (6.09e+15) =	1.5930e+17 (3.35e+14) -	1.5875e+17 (7.25e+14) -
WFG6	5	4.5654e+3 (9.90e+1)	4.2795e+3 (7.58e+1)-	4.5868e+3 (9.09e+1) =	4.5801e+3 (4.92e+1) =	4.4167e+3 (8.13e+1) -	4.4436e+3 (9.67e+1) -	4.4366e+3 (7.60e+1) =	4.4989e+3 (7.04e+1) =	4.4500e+3 (2.14e+2) =	4.5358e+3 (9.28e+1) =
	8	1.8830e+7 (4.93e+5)	1.7996e+7 (2.19e+5)-	1.8617e+7 (4.21e+5) =	1.8692e+7 (4.17e+5) =	1.8468e+7 (3.56e+5) =	1.5454e+7 (6.13e+5) -	1.8041e+7 (3.80e+5) -	1.7430e+7 (6.69e+5) -	1.8384e+7 (4.97e+5) =	1.8256e+7 (4.36e+5) -
	10	8.5473e+9 (1.88e+8)	8.3704e+9 (9.11e+8)-	8.4620e+9 (1.93e+8) =	8.5286e+9 (1.36e+8) =	8.3524e+9 (1.76e+8) -	6.8594e+9 (3.22e+8) -	8.1620e+9 (1.76e+8) -	7.9963e+9 (4.28e+8) -	8.3362e+9 (2.83e+8) -	8.3008e+9 (2.44e+8) -
	15	1.6154e+17 (4.13e+15)	1.4535e+17 (5.93e+15)-	1.5481e+17 (6.14e+15) -	1.5570e+17 (7.76e+15) -	1.5608e+17 (4.38e+15) -	9.6400e+16 (5.26e+15) -	1.5605e+17 (3.55e+15) -	1.2060e+17 (9.52e+15) -	1.5234e+17 (6.62e+15) -	1.4979e+17 (4.27e+15) -

Instances	M	MaOEA-RD	ISNPS [43]	RVEA [52]	NSGA-III [16]	VaEA [63]	MOEA/DD [17]	RVEA* [52]	ANSGA-III [50]	MOEA/D-AWA [51]	MOEA/D-URAW [55]
WFG7	5	4.4775e+3 (6.48e+1)	4.6579e+3 (5.92e+1)+	4.9914e+3 (4.50e+0) +	5.0001e+3 (4.22e+0) +	4.8514e+3 (2.00e+1) +	4.8282e+3 (1.37e+1) +	4.8548e+3 (1.30e+1) +	4.9117e+3 (2.37e+1) +	4.9966e+3 (1.75e+1) +	4.9741e+3 (8.24e+0) +
	8	2.0409e+7 (1.02e+4)	1.8297e+7 (4.36e+5)-	2.0194e+7 (6.03e+4) -	2.0286e+7 (1.75e+4) -	2.0071e+7 (5.19e+4) -	1.8328e+7 (3.68e+5) -	1.9588e+7 (9.68e+4) -	1.9135e+7 (2.84e+5) -	2.0198e+7 (3.13e+4) -	2.0069e+7 (5.20e+4) -
	10	9.3312e+9 (2.77e+6)	8.7238e+9 (1.35e+8)-	9.2939e+9 (8.07e+6) -	9.2439e+9 (2.01e+8) -	9.1241e+9 (3.36e+7) -	8.2017e+9 (8.13e+7) -	8.9554e+9 (2.93e+7) -	8.8780e+9 (4.42e+7) -	9.2453e+9 (5.97e+6) -	9.1713e+9 (1.84e+7) -
	15	1.7730e+17 (2.12e+13)	1.4980e+17 (6.94e+15)-	1.7722e+17 (4.34e+13) -	1.7549e+17 (1.23e+15) -	1.7214e+17 (3.15e+14) -	1.1127e+17 (8.33e+13) -	1.7338e+17 (4.25e+14) -	1.7073e+17 (7.07e+15) -	1.6961e+17 (4.33e+15) -	1.7097e+17 (6.81e+14) -
	5	4.3419e+3 (1.18e+1)	3.7076e+3 (7.03e+1)-	4.9914e+3 (4.50e+0) +	4.3018e+3 (1.12e+1) -	4.0365e+3 (5.21e+1) -	4.1854e+3 (5.12e+1) -	4.1758e+3 (3.22e+1) -	3.9797e+3 (5.08e+1) -	3.5988e+3 (1.72e+2) -	4.3371e+3 (1.08e+1) =
WFG8	8	1.7867e+7 (8.58e+4)	1.7116e+7 (1.69e+5)-	1.7097e+7 (1.65e+6) -	1.7812e+7 (9.82e+5) =	1.6142e+7 (2.99e+5) -	1.5729e+7 (1.49e+6) -	1.7160e+7 (1.33e+6) -	1.5695e+7 (5.81e+5) -	1.7600e+7 (1.28e+5) -	1.8090e+7 (1.30e+5) +
	10	8.4562e+9 (5.16e+7)	8.1881e+9 (1.02e+8)-	8.2951e+9 (4.98e+8) -	8.1267e+9 (3.04e+8) -	7.7684e+9 (2.38e+8) -	6.6857e+9 (7.74e+8) -	8.3877e+9 (5.43e+8) =	7.9096e+9 (1.38e+8) -	8.1965e+9 (4.30e+7) -	8.5301e+9 (2.06e+7) +
	15	1.6642e+17 (9.23e+14)	1.4697e+17 (5.45e+15)-	1.3912e+17 (2.49e+16) -	1.4955e+17 (1.67e+16) -	1.5006e+17 (1.68e+15) -	1.0674e+17 (3.22e+16) -	8.1782e+16 (7.35e+15) -	1.1511e+17 (1.66e+16) -	1.6116e+17 (2.06e+15) -	1.1619e+17 (9.03e+15) -
	5	4.7148e+3 (3.95e+1)	3.9680e+3 (2.87e+1)-	4.7396e+3 (2.08e+1) =	4.7121e+3 (2.96e+1) =	4.5133e+3 (2.73e+1) -	4.5095e+3 (3.65e+1) -	4.5063e+3 (4.47e+1) -	4.6158e+3 (4.04e+1) -	4.2606e+3 (3.51e+2) -	4.5902e+3 (2.16e+2) -
	8	1.7737e+7 (1.76e+6)	1.5635e+7 (2.49e+5)-	1.8963e+7 (4.94e+5) =	1.8764e+7 (5.57e+5) =	1.7978e+7 (8.37e+5) =	1.6202e+7 (7.35e+5) =	1.7699e+7 (2.32e+5) =	1.6839e+7 (1.41e+6) -	1.8027e+7 (1.18e+6) =	1.7560e+7 (1.05e+6) =
WFG9	10	8.4059e+9 (6.08e+8)	7.1801e+9 (9.03e+8)-	8.8440e+9 (8.58e+7) +	8.4967e+9 (7.02e+7) =	8.1178e+9 (3.74e+8) -	6.9224e+9 (2.35e+8) -	7.9890e+9 (1.45e+8) -	8.0826e+9 (4.79e+8) -	7.9800e+9 (7.62e+8) -	8.0286e+9 (3.97e+8) -
	15	1.5723e+17 (1.65e+16)	1.2398e+17 (4.51e+15)-	1.5744e+17 (1.03e+16) =	1.4483e+17 (1.23e+16) -	1.4483e+17 (1.23e+16) -	9.5702e+16 (1.06e+16) -	1.0844e+17 (1.43e+16) -	1.4325e+17 (9.82e+15) -	1.4759e+17 (1.13e+16) -	1.3634e+17 (6.84e+15) -
	+/-/=	-	-/-	3/32/1	7/17/12	8/18/10	8/24/4	5/27/5	1/32/3	7/25/4	4/26/6

To visualize the efficiency of MaOEA-RD and all the compared algorithms, some results of the parallel coordinates and the convergence<sup>11</sup> plots on the 10-dimensional problems are presented. As can be seen from Figure 14 to Figure 21, MaOEA-RD, RVEA\*, MOEA/D-AWA and MOEA/D-URAW have good convergence, while RVEA and NSGA-III are not convergent on some problems.



<sup>11</sup>Convergence= $\frac{\| \{F(X) \in PF, \exists F(X^*) \in PF^*, F(X^*) \prec F(X)\} \|}{\|PF\|}$ , where the  $PF$  is achieved by the algorithm and the  $PF^*$  is the  $PF$ . The smaller the convergence value, the better the result. More details can be found in [27].

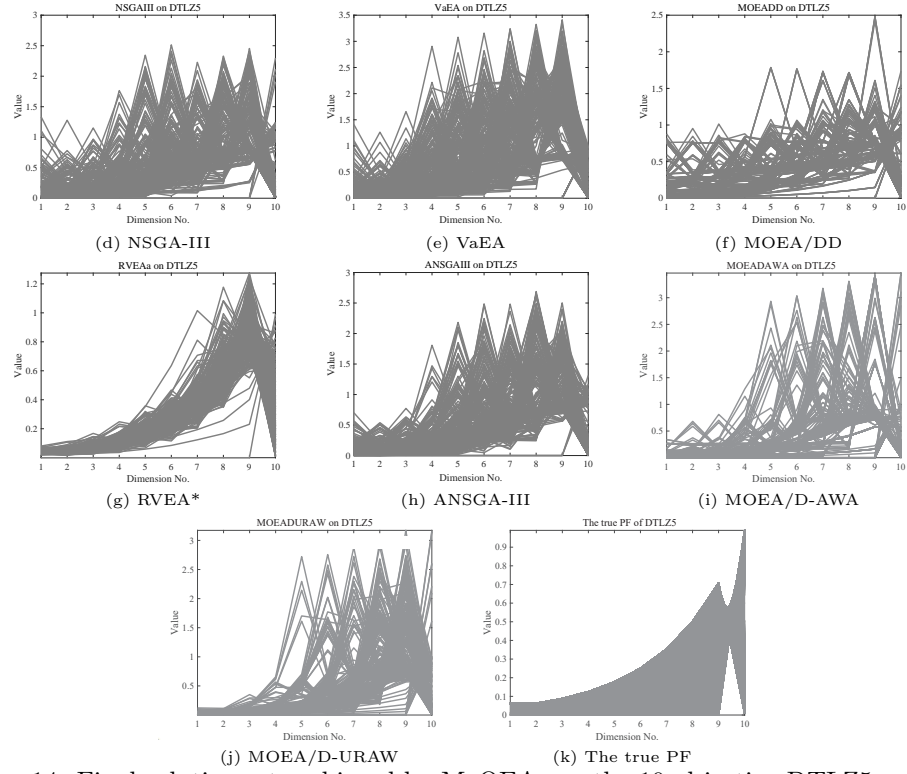
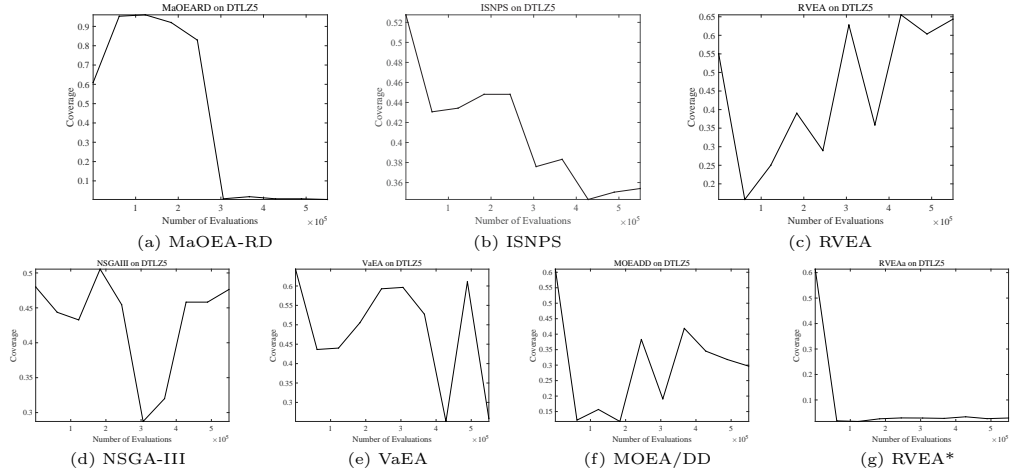


Figure 14: Final solution sets achieved by MaOEAs on the 10-objective DTLZ5 problem, shown by parallel coordinates.



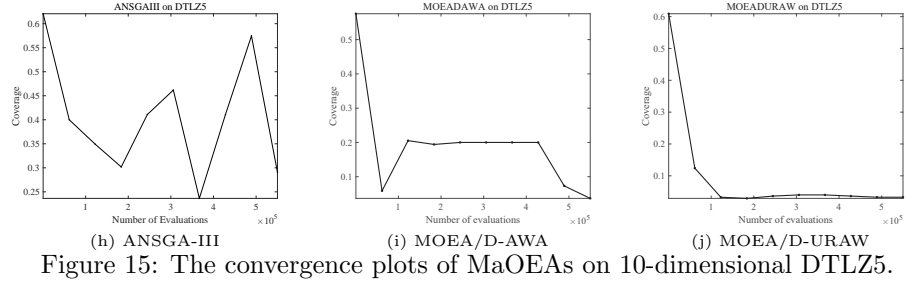


Figure 15: The convergence plots of MaOEAs on 10-dimensional DTLZ5.

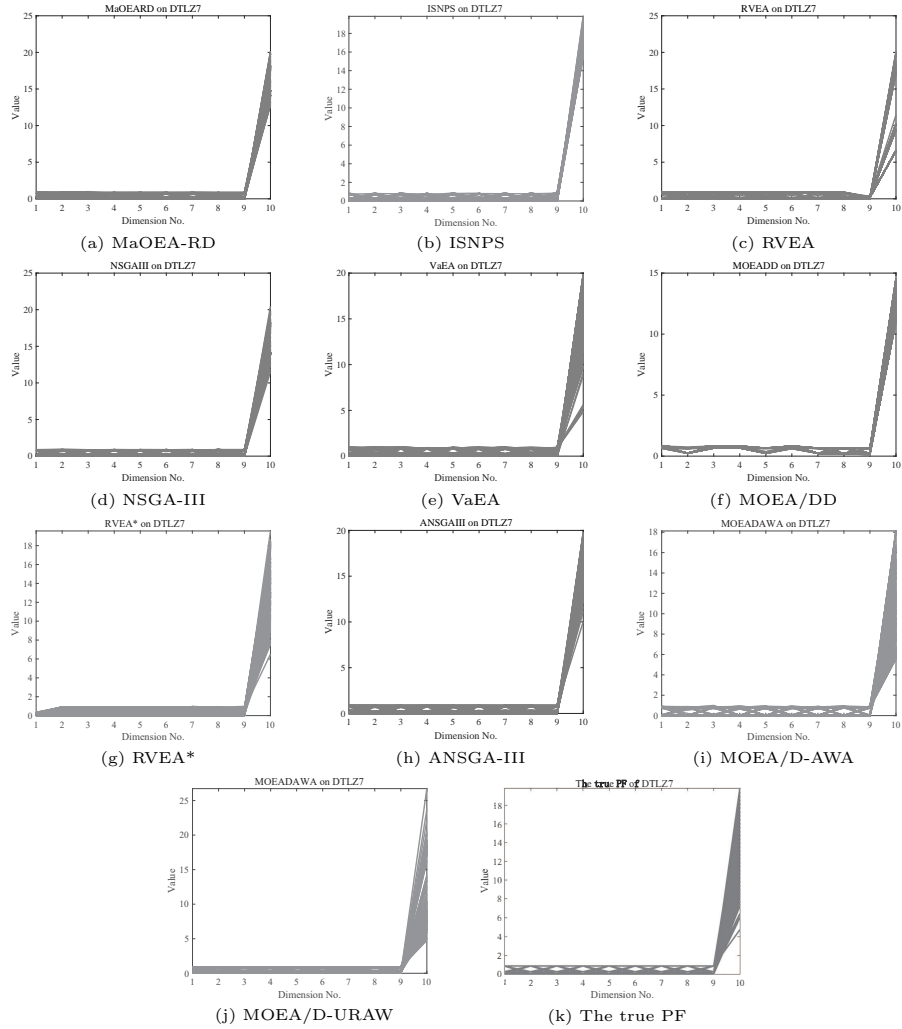


Figure 16: Final solution sets achieved by MaOEAs on the 10-objective DTLZ7 problem, shown by parallel coordinates.

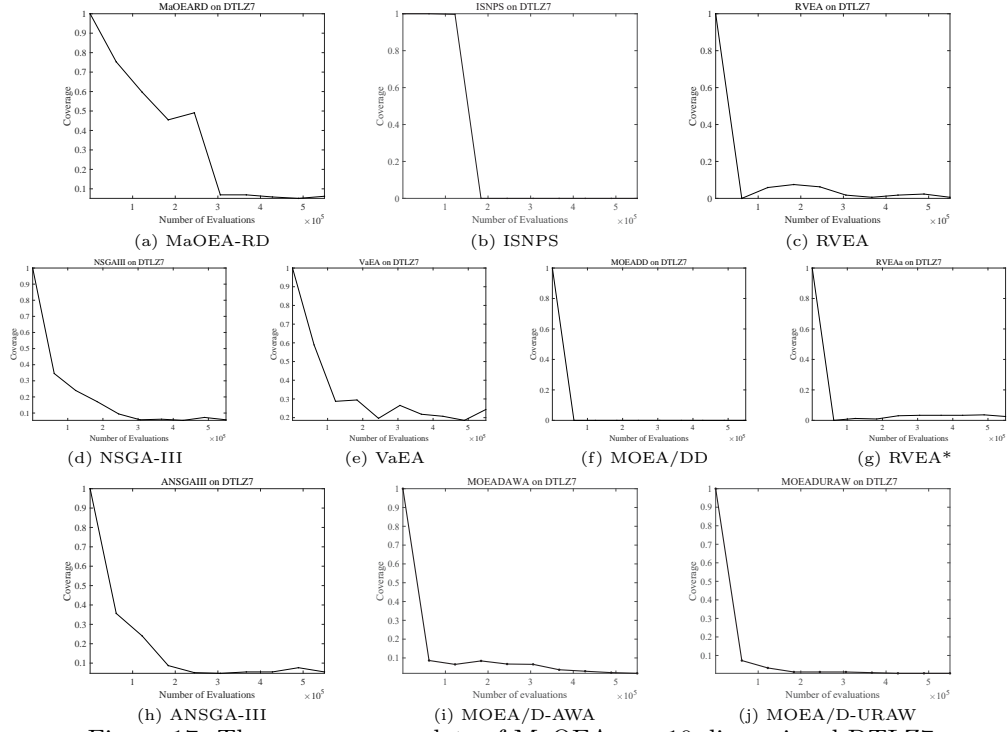
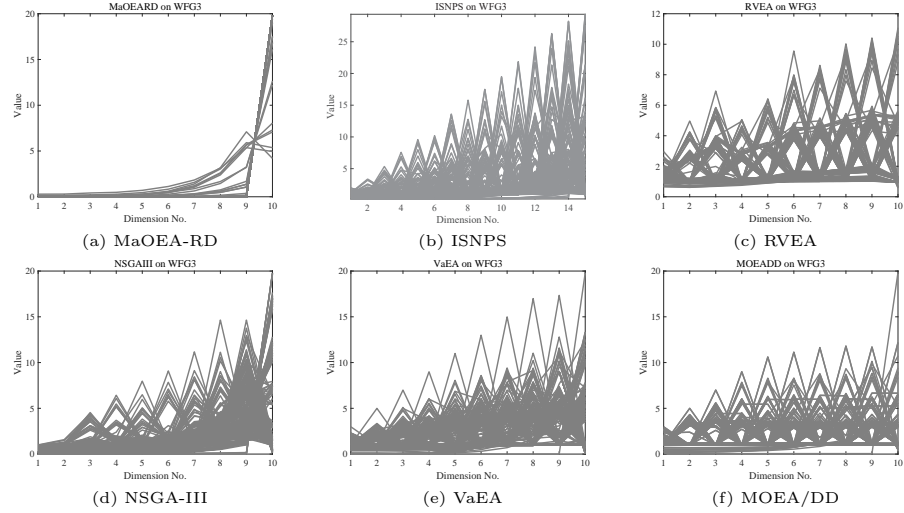


Figure 17: The convergence plots of MaOEAs on 10-dimensional DTLZ7.



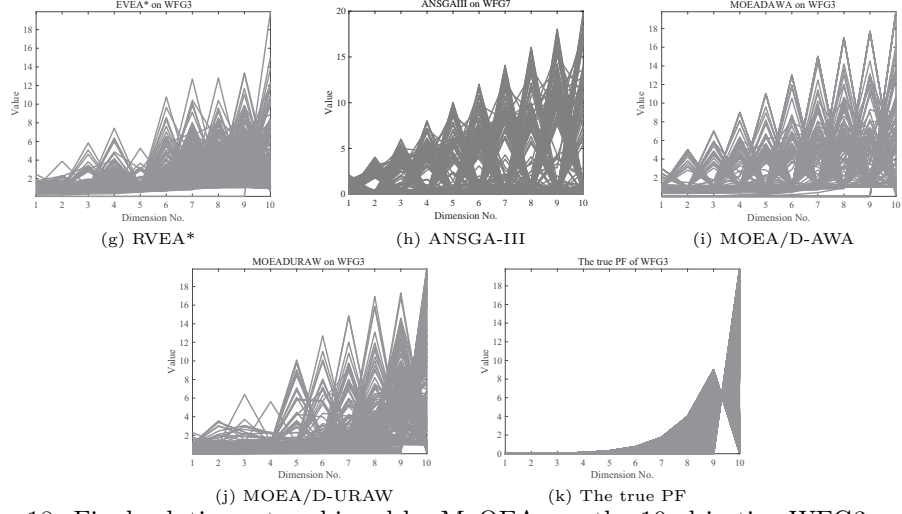


Figure 18: Final solution sets achieved by MaOEAs on the 10-objective WFG3 problem, shown by parallel coordinates.

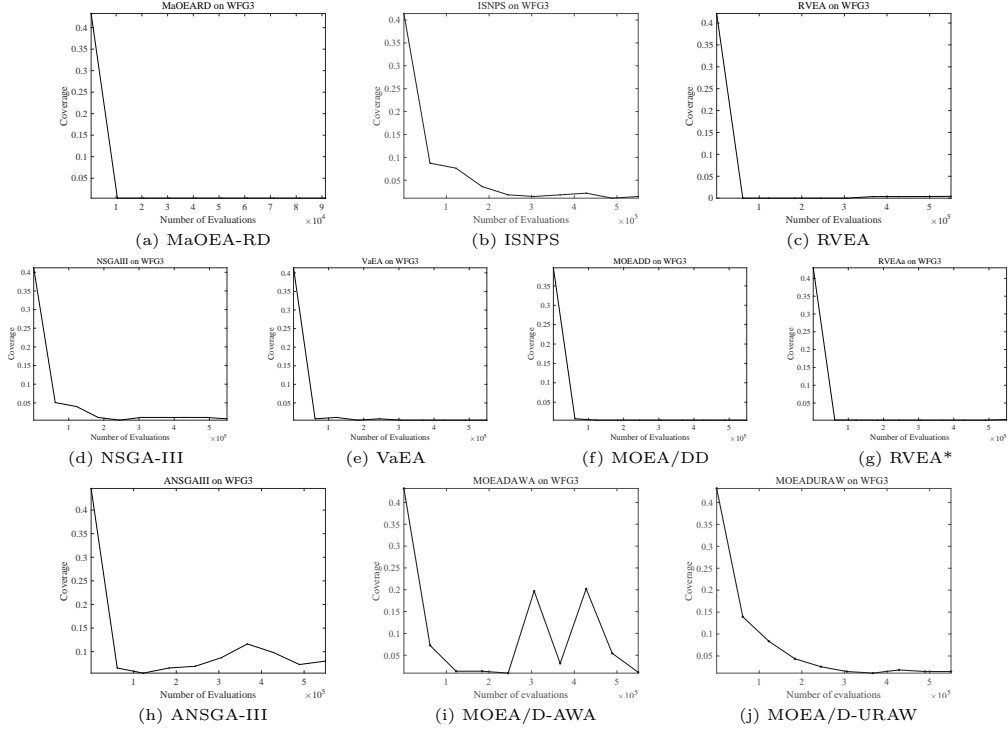


Figure 19: The convergence plots of MaOEAs on 10-dimensional WFG3.

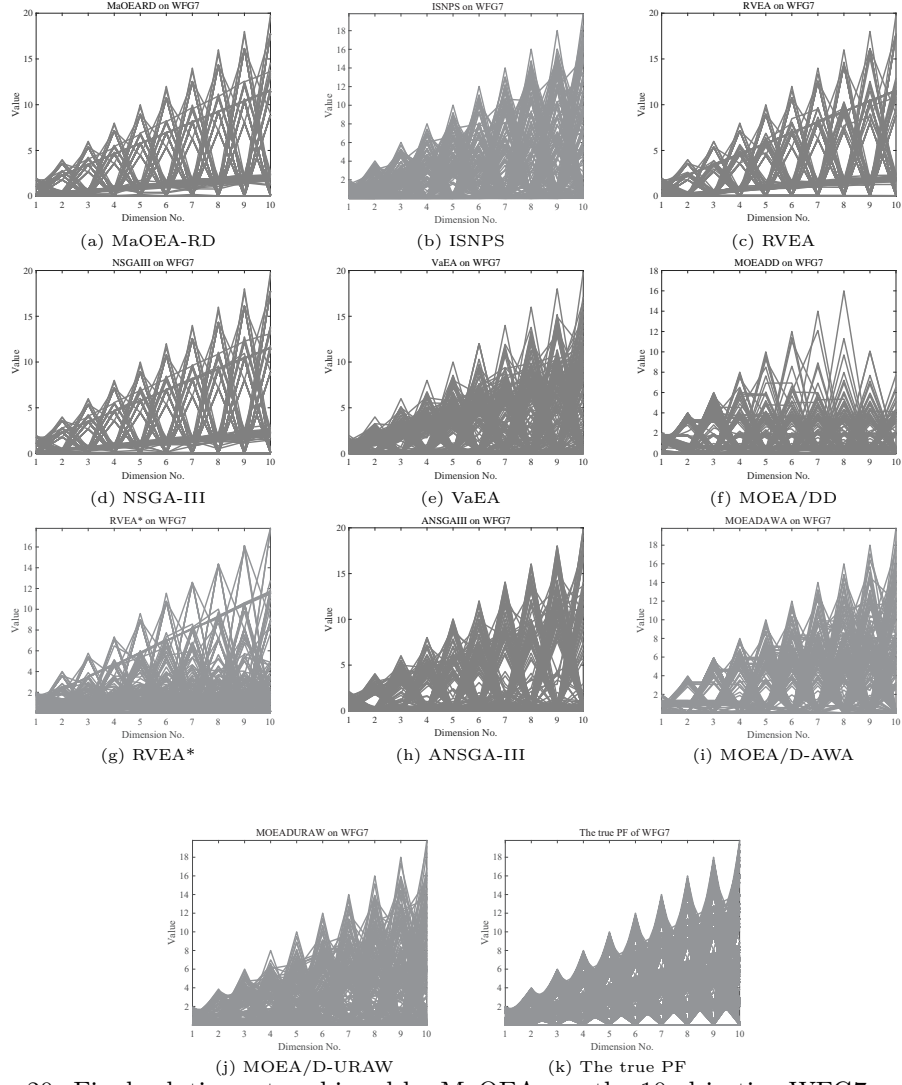
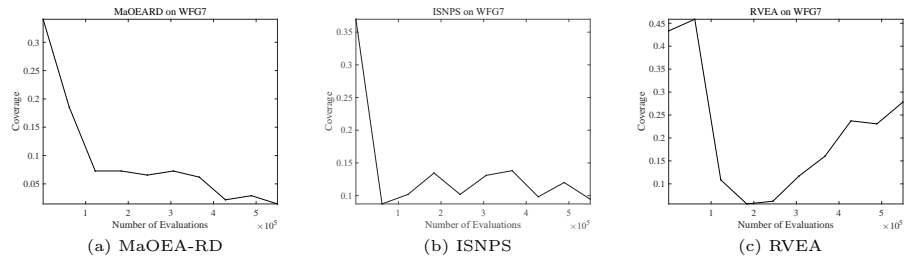


Figure 20: Final solution sets achieved by MaOEAs on the 10-objective WFG7 problem, shown by parallel coordinates.





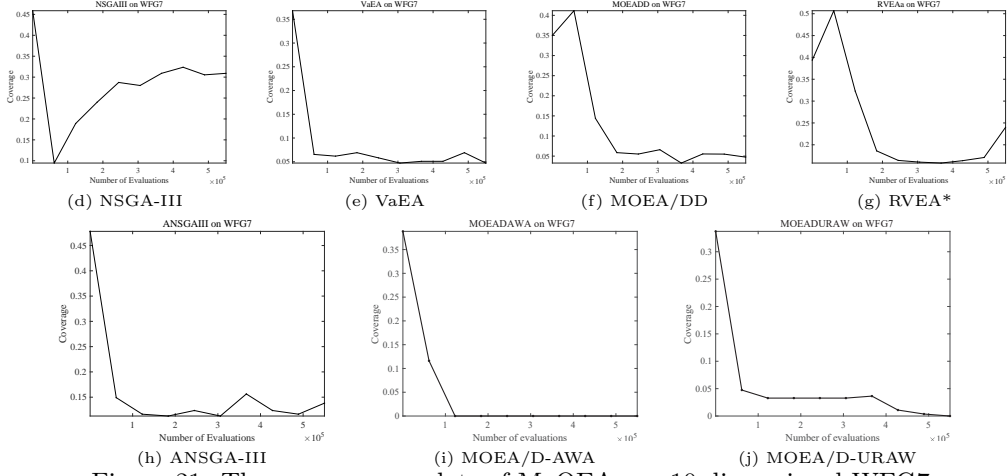


Figure 21: The convergence plots of MaOEAs on 10-dimensional WFG7.

#### 4.4. Further investigations of MaOEA-RD

In this subsection, we will discuss several parameters that can affect the performance of MaOEA-RD. They are the timing of the adjustment of the reference vectors ( $\varphi_1$ ), the *CI* and the vector adjustment mechanism ( $\varphi_2$  and  $\varphi_3$ ).

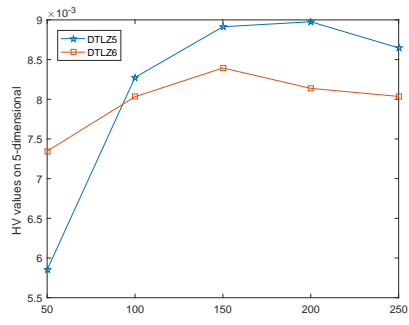
To detect the influence of the vector adjustment on algorithm performance, we conducted experiments<sup>12</sup> on DTLZ5 and DTLZ6 to adjust the reference vectors<sup>13</sup> every 50, 100, 150, 200 and 250 iterations, and drew the HV values into a broken line graph as shown in Figure 23. As can be seen, the best HV values are obtained by adjusting the reference vectors every 150 generations<sup>14</sup>. This is because the frequency of the reference vectors' adjustment can affect the performance of the algorithm. If the reference vectors are adjusted too frequently, the population cannot evolve properly; inversely, if the reference vectors are adjusted too slowly, they have little opportunity to adjust throughout the evolution.

In order to overcome the shortcoming that adjusting the reference vectors may compromise the performance of the algorithm on regular PFs, the *CI* is adopted as a switch that determines whether the vectors are adjusted or

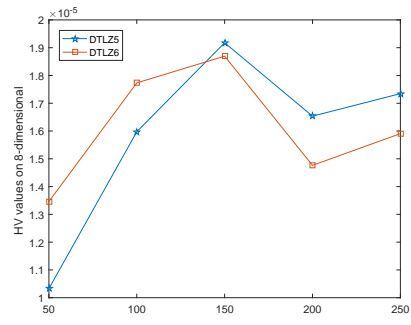
<sup>12</sup>All parameters settings are consistent with those mentioned previously.

<sup>13</sup>To detect the effect of frequency on performance,  $\varphi_2$  and  $\varphi_3$  are set to  $-\infty$  and  $+\infty$ , respectively.

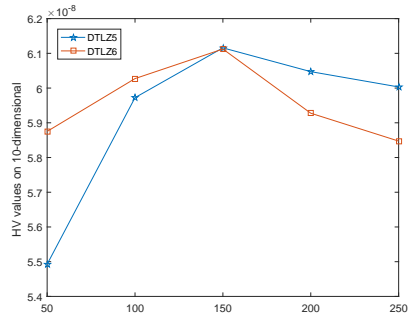
<sup>14</sup>We recommend that the  $\varphi_1$  is set to 150.



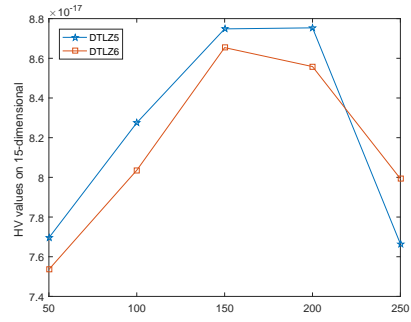
(a)



(b)



(c)



(d)

Figure 22: The HV results of adjusting reference vectors every 50, 100, 150, 200 and 150 generations.

not. In Algorithm 4, the evolutionary potential of the population can be reflected by the ratio of  $CI$  to  $HisCI$ . In other words, the closer the ratio is to 1, the smaller the evolutionary potential of the population. We conducted experiments<sup>15</sup> on DTLZ2 and DTLZ4 to detect the influence of  $\varphi_2$  and  $\varphi_3$ , and drew the HV values into broken line graphs as shown in Figure 23. As can be seen from the figure, when  $\varphi_2$  is between  $[0.85, 1]$  and  $\varphi_3$  is between  $[1, 1.15]$ , the HV values are relatively large<sup>16</sup>.

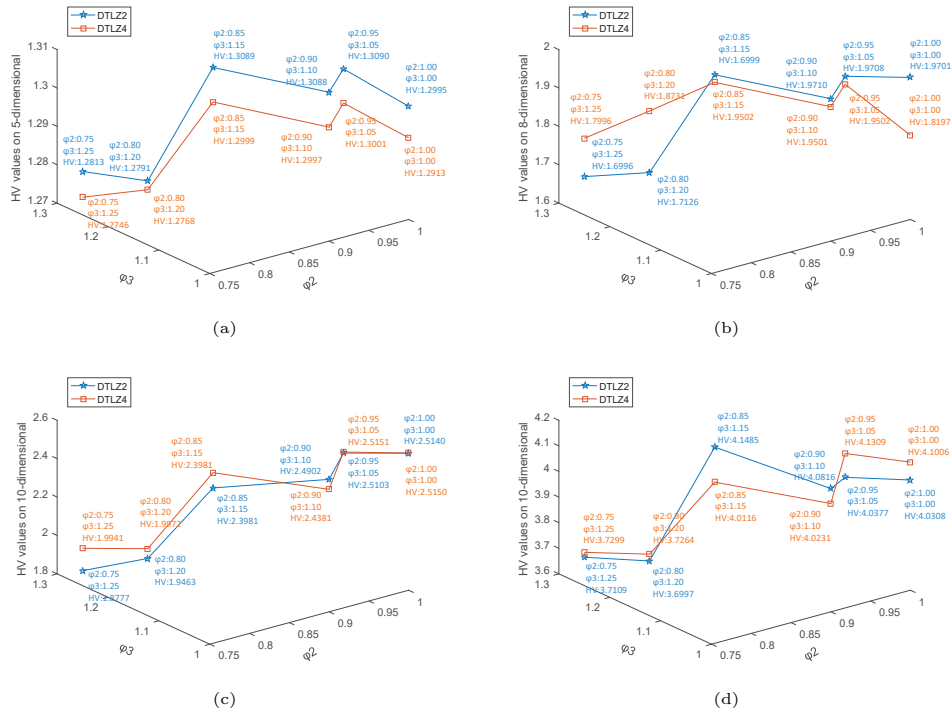


Figure 23: The HV results with different  $\varphi_2$  and  $\varphi_3$ .

To explore the role of the  $CI$  and the vector adjustment mechanism, we carry out three variants of MaOEA-RD. For the first variant, the  $PBI$  in [7] is used to replace  $CI$  and  $PBI_m$ ; for the second variant, there is no vector adjustment; and for the third variant, the mechanism proposed in this paper is deleted (part one of Algorithm 4), but the vectors are adjusted every 150 generations. All parameters settings are the same as those mentioned.

<sup>15</sup>The  $\varphi_1$  is set to 150 in this experimental design.

<sup>16</sup>We recommend that the  $\varphi_2$  is set to 0.95 and the  $\varphi_3$  is set to 1.15.

MaOEA-RD and the three variants are run on DTLZ and WFG problems, and the HV results are shown in Table 7 and Table 8, respectively.

Table 7: Mean and standard deviation of HV values on DTLZ problems of MaOEA-RD and its variants

Instances	M	MaOEA-RD	Variant1	Variant2	Variant3
DTLZ1	5	4.9316e-2 (6.17e-6)	4.9305e-2 (1.65e-5) -	4.9314e-2 (2.21e-6)-	4.9304e-2 (2.45e-5)-
	8	8.3532e-3 (5.05e-7)	8.3532e-3 (4.29e-7) =	8.3533e-3 (6.28e-7)=	8.3528e-3 (1.03e-6)-
	10	2.5322e-3 (2.77e-8)	2.5321e-3 (4.18e-8) -	2.5322e-3 (1.00e-8)=	2.5318e-3 (6.85e-7)-
	15	1.2747e-4 (2.10e-9)	1.1568e-4 (1.21e-5) -	1.2747e-4 (1.74e-9)=	1.2703e-4 (2.24e-7)-
DTLZ2	5	1.3091e+0 (4.88e-4)	1.3083e+0 (5.96e-4) -	1.3082e+0 (5.55e-4)-	1.3078e+0 (5.69e-4)-
	8	1.9797e+0 (6.16e-4)	1.9793e+0 (5.06e-4) =	1.9799e+0 (5.94e-4)=	1.9797e+0 (5.77e-4)=
	10	2.5150e+0 (3.38e-4)	2.5148e+0 (5.06e-4) =	2.5148e+0 (3.46e-4)-	2.5148e+0 (5.27e-4)+
	15	4.1377e+0 (5.46e-4)	4.1372e+0 (1.64e-3) =	4.1383e+0 (5.12e-4)=	4.1111e+0 (1.86e-2)-
DTLZ3	5	1.3061e+0 (1.66e-3)	1.2978e+0 (9.60e-3) -	1.3039e+0 (1.84e-3)-	1.2095e+0 (2.61e-2)-
	8	1.9738e+0 (6.29e-3)	1.9734e+0 (4.08e-3) =	1.9743e+0 (3.49e-3)=	1.9103e+0 (3.34e-2)-
	10	2.5147e+0 (9.59e-4)	2.5135e+0 (1.43e-3) -	2.5140e+0 (9.69e-4)=	2.4852e+0 (1.62e-2)=
	15	4.1143e+0 (8.75e-2)	4.1371e+0 (1.34e-3) =	4.1383e+0 (6.00e-4)+	3.8694e+0 (8.52e-2)-
DTLZ4	5	1.3085e+0 (5.56e-4)	1.2976e+0 (4.23e-2) =	1.3081e+0 (5.16e-4)-	1.3084e+0 (5.86e-4)=
	8	1.9508e+0 (4.73e-2)	1.9716e+0 (2.31e-2) =	1.9630e+0 (3.86e-2)=	1.9716e+0 (2.33e-2)+
	10	2.5151e+0 (2.44e-4)	2.5152e+0 (4.09e-4) =	2.5154e+0 (3.85e-4)=	2.5150e+0 (4.24e-4)=
	15	4.1306e+0 (1.05e-2)	4.1339e+0 (6.97e-3) =	4.1292e+0 (1.98e-2)-	4.1455e+0 (4.86e-4)+

Instances	M	MaOEA-RD	Variant1	Variant2	Variant3	
DTLZ5	5	8.9138e-3	8.8709e-3	8.1664e-3	8.9717e-3	
		(1.55e-4)	(3.22e-4) -	(2.09e-5)-	(7.21e-5)+	
	8	1.9172e-5	1.9013e-5	1.7733e-5	1.9354e-5	
		(2.95e-7)	(1.30e-7) -	(2.17e-8)-	(1.33e-7)=	
	10	6.1584e-8	6.1295e-8	5.8313e-8	6.1781e-8	
		(5.21e-10)	(9.72e-10) -	(2.17e-10)-	(4.52e-10)+	
	15	8.7480e-17	8.7323e-17	8.5492e-17	8.7881e-17	
		(6.36e-19)	(6.31e-19) =	(1.13e-19)-	(5.61e-19)+	
	DTLZ6	5	8.3943e-3	8.3141e-3	8.1837e-3	9.1098e-3
			(2.20e-4)	(3.38e-4) -	(2.21e-5)-	(3.40e-5)+
		8	1.8696e-5	1.8002e-5	1.7702e-5	1.9432e-5
			(5.39e-7)	(2.61e-7) -	(3.73e-8)-	(8.29e-8)+
10		6.0100e-8	6.0435e-8	5.8208e-8	6.1559e-8	
		(1.19e-9)	(1.24e-9) =	(3.14e-10)-	(2.79e-10)+	
15		8.6570e-17	8.4533e-17	8.5670e-17	8.7766e-17	
		(1.06e-18)	(1.01e-18) -	(2.91e-19)-	(5.40e-19)+	
DTLZ7		5	2.2401e+0	1.1229e+0	2.1850e+0	2.3243e+0
			(1.37e-1)	(1.97e-1) -	(7.33e-2)-	(3.71e-2)+
		8	2.2911e+0	7.7471e-1	2.2933e+0	2.4534e+0
			(7.47e-2)	(7.46e-1)-	(9.74e-2)=	(3.56e-2)
	10	2.3647e+0	1.8837e-1	2.3020e+0	2.6425e+0	
		(5.82e-2)	(3.35e-1) -	(6.56e-2)-	(4.45e-2)+	
	15	2.3194e+0	3.8336e-2	2.3696e+0	2.3493e+0	
		(6.94e-2)	(6.29e-2) -	(1.52e-1)+	(3.88e-2)+	
	+/-/=		0/16/12	2/16/10	14/9/5	

First, comparing MaOEA-RD with its variant1, we find that MaOEA-RD performs the best 30 times out of the 64 problems and there is no significant difference between the two in 22 problems. This means that using the *CI* and *PBI<sub>m</sub>* is more likely to lead to better population evolution than using the *PBI*. This is because using the *CI* as the first selection criterion is more selective than the *PBI* approach, so there are more opportunities to activate the reference vector adjustment mechanism to maintain good diversity.

Secondly, an interesting phenomenon can be seen by comparing variant2 and variant3. The results of adjusting the reference vectors are not as good as those without adjusting the reference vectors for problems with regular

PFs (DTLZ1-DTLZ4). In contrast, for problems with irregular PFs, the variant3 has better results (DTLZ5-DTLZ7 and WFG3). This phenomenon is consistent with Giagkiozis et al. [58]. One conclusion that can be drawn is that the reference vector adjustment provides a significant performance improvement for problems with irregular PFs, but is counterproductive for problems with regular PFs.

Table 8: Mean and standard deviation of HV values on WFG problems of MaOEA-RD and its variants

Instances	M	MaOEA-RD	Variant1	Variant2	Variant3
WFG1	5	6.0423e+3 (5.54e-1)	6.0418e+3 (6.44e-1) -	6.0427e+3 (5.84e-1)+	6.0345e+3 (2.97e+0)-
	8	2.0258e+7 (9.03e+5)	2.0693e+7 (3.04e+4) =	2.0729e+7 (2.24e+3)+	2.0198e+7 (2.26e+4)-
	10	8.5436e+9 (2.81e+8)	8.6485e+9 (5.06e+6) =	8.6635e+9 (8.65e+5)+	8.4548e+9 (2.43e+6)-
	15	1.3904e+17 (7.85e+13)	1.3853e+17 (1.30e+14) -	1.3913e+17 (2.84e+12)+	1.3896e+17 (7.99e+13)-
WFG2	5	6.1623e+3 (4.13e+0)	6.1591e+3 (3.62e+0) =	6.1629e+3 (3.31e+0)=	6.1635e+3 (2.26e+0)=
	8	2.1122e+7 (3.88e+5)	2.1904e+7 (4.15e+4) +	2.1744e+7 (6.69e+5)+	2.1119e+7 (5.16e+5)=
	10	9.1141e+9 (4.76e+8)	9.4898e+9 (3.31e+7) +	9.5313e+9 (6.15e+7)+	9.0871e+9 (3.98e+8)-
	15	1.4650e+17 (2.43e+16)	1.6749e+17 (2.15e+15) =	1.6481e+17 (8.40e+15)+	1.0450e+17 (5.47e+16)-
WFG3	5	2.9256e+0 (2.02e-1)	2.9424e+0 (1.98e-1)=	2.7556e+0 (2.83e-0)-	2.8749e+0 (2.55e-1)-
	8	6.3310e-3 (5.72e-3)	1.7117e-3 (2.98e-3) -	2.4820e-2 (1.98e-3)+	2.6316e-2 (1.42e-3)+
	10	7.0402e-6 (1.52e-5)	0.0000e+0 (0.00e+0) -	6.2553e-5 (1.06e-5)+	7.8005e-5 (6.32e-6)+
	15	5.0764e-20 (1.97e-19)	5.2239e-17 (1.29e-16) =	6.4941e-18 (1.45e-17)+	6.1249e-17 (8.40e-17)+

Instances	M	MaOEA-RD	Variant1	Variant2	Variant3
WFG4	5	4.9931e+3 (6.05e+0)	4.9895e+3 (4.75e+0) =	5.0036e+3 (5.37e+0)=	4.9856e+3 (5.10e+0)-
	8	2.0414e+7 (2.10e+4)	1.9141e+7 (2.06e+5) -	2.0422e+7 (1.16e+4)=	2.0426e+7 (7.24e+3)=
	10	9.3372e+9 (8.05e+6)	8.8653e+9 (6.12e+7) -	9.3233e+9 (6.38e+6)-	9.3275e+9 (4.03e+6)-
	15	1.7732e+17 (2.03e+13)	1.7339e+17 (1.87e+15) -	1.7729e+17 (2.04e+13)-	1.7674e+17 (5.38e+14)-
WFG5	5	4.7085e+3 (1.80e+0)	4.7080e+3 (2.55e+0) =	4.7094e+3 (4.93e+0)=	4.7072e+3 (8.46e-1)-
	8	1.9110e+7 (4.64e+3)	1.7742e+7 (2.41e+5) -	1.9109e+7 (2.62e+3)-	1.9108e+7 (6.48e+3)-
	10	8.7216e+9 (1.48e+6)	8.1268e+9 (8.18e+7) -	8.7188e+9 (2.01e+6)-	8.7174e+9 (1.54e+6)-
	15	1.6424e+17 (1.68e+13)	1.4816e+17 (7.64e+15) -	1.6423e+17 (1.26e+13)=	1.6422e+17 (1.67e+13)-
WFG6	5	4.5654e+3 (9.90e+1)	4.6507e+3 (1.28e+2) +	4.6711e+3 (6.39e+1)+	4.6133e+3 (9.04e+1)+
	8	1.8830e+7 (4.93e+5)	1.6899e+7 (5.40e+5) -	1.8626e+7 (1.91e+5)-	1.9340e+7 (3.81e+5)+
	10	8.5473e+9 (1.88e+8)	7.7897e+9 (2.09e+8) -	8.4872e+9 (2.69e+8)-	8.6266e+9 (1.63e+8)+
	15	1.6154e+17 (4.13e+15)	1.2309e+17 (2.08e+16) -	1.6051e+17 (3.01e+15)-	1.5995e+17 (6.47e+15)-
WFG7	5	4.4775e+3 (6.48e+1)	5.0026e+3 (3.94e+0) +	5.0039e+3 (3.68e+0)+	5.0034e+3 (2.15e+0)+
	8	2.0409e+7 (1.02e+4)	1.9066e+7 (1.63e+5) -	2.0392e+7 (7.20e+3)-	2.0395e+7 (1.50e+4)-
	10	9.3312e+9 (2.77e+6)	8.9317e+9 (4.84e+7) -	9.3233e+9 (3.01e+6)-	9.3229e+9 (2.63e+6)-
	15	1.7730e+17 (2.12e+13)	1.7439e+17 (1.02e+15) -	1.7729e+17 (2.41e+13)-	1.7658e+17 (2.71e+14)-
WFG8	5	4.3419e+3 (1.18e+1)	4.3229e+3 (1.41e+1) -	4.3484e+3 (1.23e+1)=	4.3335e+3 (8.62e+0)-
	8	1.7867e+7 (8.58e+4)	1.7191e+7 (4.74e+5) -	1.7803e+7 (9.04e+4)-	1.7933e+7 (5.36e+4)+
	10	8.4562e+9 (5.16e+17)	7.8784e+9 (5.39e+8) -	8.4246e+9 (1.61e+7)-	8.4819e+9 (1.51e+7)+
	15	1.6642e+17 (9.23e+14)	1.5497e+17 (1.06e+16) -	1.6455e+17 (1.83e+15)-	1.6561e+17 (1.21e+15)-
WFG9	5	4.7148e+3 (3.95e+1)	4.7386e+3 (2.94e+1) =	4.7578e+3 (2.26e+1)+	4.7323e+3 (2.40e+1)+
	8	1.7737e+7 (1.76e+6)	1.6712e+7 (4.69e+5) =	1.8590e+7 (1.69e+6)+	1.7731e+7 (1.96e+6)=
	10	8.4059e+9 (6.08e+8)	7.4036e+9 (6.31e+8) -	8.8421e+9 (3.01e+7)+	8.4858e+9 (7.72e+8)-
	15	1.5723e+17 (1.65e+16)	1.1080e+17 (1.80e+16) -	1.6022e+17 (1.64e+16)+	1.6447e+17 (3.02e+15)+
+/-/=			4/14/10	16/7/5	9/15/4

Thirdly, comparing MaOEA-RD with its variant2 and variant3, one intuitive finding is that MaOEA-RD is a compromise between variant2 and variant3. Since MaOEA-RD restores the reference vectors to the optimal historical record if the adjustment of the reference vectors fails to work, it can avoid the influence of the adjustment of the reference vectors on problems with regular PFs. In contrast, when facing problems with irregular PFs, MaOEA-RD can inherit the advantages of variant3. As can be seen from Table 7, variant2 performs better than variant3 on DTLZ1-DTLZ4 while variant3 performs better than variant2 on DTLZ5-DTLZ6. As can be seen from Table 8, the performance of MaOEA-RD is not as good as its variant2 in WFG1-WFG3 and WFG9 problems. This may be due to a mix of features on these issues.

#### 4.5. Overall performance

Same as ISNPS, our algorithm enhances the selection pressure of individuals by the  $CI$  values. To maintain a good distribution of the population, MaOEA-RD combines the reference vectors and proposes a vector adjustment mechanism. One thing that is clear is that an algorithm cannot solve all problems with different properties. ISNPS, RVEA (and RVEA\*), NSGA-III (and ANSGA-III), VAEA, MOEA/DD, MOEA/D-AWA and MOEA/D-URAW are excellent algorithms and they have their own strengths on different problems. Compared with them, MaOEA-RD is competitive.

### 5. Summary

In this paper, we have proposed a many-objective optimization algorithm based on rotation and decomposition. From the experimental results, we know the MaOEA-RD is very competitive. The reasons can be summarized as follows: First, MaOEA-RD uses the strength of the  $CI$  proposed by Shen et al. [43]. to select individuals with good convergence. Second, a novel environmental selection incorporating a decomposition-based method is adopted, which can maintain diversity of the population. Third, the reference vector adjustment mechanism makes MaOEA-RD compatible with more problems.

In our algorithm, the reference vector adjustment mechanism actually compromises some of the performance as mentioned in subsection 4.4. In the future, we will further investigate the reference vector adjustment mechanism, and possibly use a more flexible method to activate the reference vector adjustment to enhance the performance of our algorithm.



## 6. Acknowledgment

The authors wish to thank the support of the National Natural Science Foundation of China (Grant No. 61876164, 61502408, 61673331), the Education Department Major Project of Hunan Province (Grant No. 17A212), The MOEA Key Laboratory of Intelligent Computing and Information Processing, the Science and Technology Plan Project of Hunan Province (Grant No. 2016TP1020), the Provinces and Cities Joint Foundation Project (Grant No. 2017JJ4001), the Hunan province science and technology project funds (2018TP1036).

## References

- [1] I. Giagkiozis, R. C. Purshouse, P. J. Fleming, An overview of population-based algorithms for multi-objective optimisation, *International Journal of Systems Science* 46 (9) (2015) 1572–1599.
- [2] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, *IEEE Transactions on Evolutionary Computation* 6 (2) (2002) 182–197.
- [3] E. Zitzler, M. Laumanns, L. Thiele, Improving the strength pareto evolutionary algorithm, *EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems* (2000) 95–100.
- [4] M. Farina, P. Amato, On the optimal solution definition for many-criteria optimization problems, in: *Fuzzy Information Processing Society, 2002. Proceedings. Nafips. 2002 Meeting of the North American*, 2002, pp. 233–238.
- [5] Y. C. Zhihua Cui, X. C. Jiangjiang Zhang, W. Zhang, Improved nsga-iii with selection-and-elimination operator, *Swarm and Evolutionary Computation* 49 (2019) 23–33.
- [6] D. Hadka, P. Reed, Borg: An auto-adaptive many-objective evolutionary computing framework, *Evolutionary computation* 21 (2) (2013) 231–259.

- [7] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, *IEEE Transactions on Evolutionary Computation* 17 (5) (2013) 721–736.
- [8] Y. Tian, R. Cheng, X. Zhang, Y. Su, Y. Jin, A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization, *IEEE Transactions on Evolutionary Computation* 23 (2) (2018) 331–345.
- [9] X. Cai, H. Sun, Q. Zhang, Y. Huang, A grid weighted sum pareto local search for combinatorial multi and many-objective optimization, *IEEE transactions on cybernetics* 49 (9) (2018) 3586–3598.
- [10] J. Zou, L. Fu, J. Zheng, S. Yang, G. Yu, Y. Hu, A many-objective evolutionary algorithm based on rotated grid, *Applied soft computing* 67 (2018) 596–609.
- [11] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, Y.-S. Ong, A new decomposition-based nsga-ii for many-objective optimization, *IEEE transactions on systems, man, and cybernetics: systems* 48 (7) (2017) 1191–1210.
- [12] Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization, *IEEE Transactions on Evolutionary Computation* 20 (1) (2015) 16–37.
- [13] Z. He, G. G. Yen, J. Zhang, Fuzzy-based pareto optimality for many-objective evolutionary algorithms, *IEEE Transactions on Evolutionary Computation* 18 (2) (2014) 269–285.
- [14] S. Kukkonen, J. Lampinen, Ranking-dominance and many-objective optimization, in: *2007 IEEE Congress on Evolutionary Computation*, IEEE, 2007, pp. 3983–3990.
- [15] Q. Zhang, L. Hui, Moea/d: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation* 11 (6) (2007) 712–731.
- [16] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i:

- Solving problems with box constraints, *IEEE Transactions on Evolutionary Computation* 18 (4) (2014) 577–601.
- [17] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, *IEEE Transactions on Evolutionary Computation* 19 (5) (2015) 694–716.
  - [18] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization by nsga-ii and moea/d with large populations, in: 2009 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2009, pp. 1758–1763.
  - [19] H. L. Liu, F. Gu, Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, *IEEE Transactions on Evolutionary Computation* 18 (3) (2014) 450–455.
  - [20] K. Sindhya, K. Miettinen, K. Deb, A hybrid framework for evolutionary multi-objective optimization, *IEEE Transactions on Evolutionary Computation* 17 (4) (2013) 495–511.
  - [21] K. Li, Q. Zhang, S. Kwong, M. Li, R. Wang, Stable matching based selection in evolutionary multiobjective optimization, *IEEE Transactions on Evolutionary Computation* 18 (6) (2014) 909–923.
  - [22] M. Wu, K. Li, S. Kwong, Q. Zhang, Evolutionary many-objective optimization based on adversarial decomposition, *IEEE transactions on cybernetics*.
  - [23] S. S. Das, M. M. Islam, N. A. Arafat, Evolutionary algorithm using adaptive fuzzy dominance and reference point for many-objective optimization, *Swarm and Evolutionary Computation* 44 (2019) 1092 – 1107.
  - [24] H. Wang, Y. Jin, X. Yao, Diversity assessment in many-objective optimization, *IEEE transactions on cybernetics* 47 (6) (2016) 1510–1522.
  - [25] X. Cai, H. Sun, Z. Fan, A diversity indicator based on reference vectors for many-objective optimization, *Information Sciences* 430 (2017) 467–486.
  - [26] Y. Tian, R. Cheng, X. Zhang, M. Li, Y. Jin, Diversity assessment of multi-objective evolutionary algorithms: Performance metric and benchmark problems, *IEEE Computational Intelligence Magazine*.

- [27] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, *IEEE transactions on Evolutionary Computation* 3 (4) (1999) 257–271.
- [28] K. Deb, S. Jain, Running performance metrics for evolutionary multi-objective optimization.
- [29] L. While, P. Hingston, L. Barone, S. Huband, A faster algorithm for calculating hypervolume, *IEEE transactions on evolutionary computation* 10 (1) (2006) 29–38.
- [30] J. Bader, E. Zitzler, Hype: an algorithm for fast hypervolume-based many-objective optimization, *Evolutionary Computation* 19 (1) (2011) 45–76.
- [31] R. H. Gómez, C. A. C. Coello, Mombi: A new metaheuristic for many-objective optimization based on the r2 indicator, in: *2013 IEEE Congress on Evolutionary Computation*, IEEE, 2013, pp. 2488–2495.
- [32] Y. Sun, G. G. Yen, Z. Yi, Igd indicator-based evolutionary algorithm for many-objective optimization problems, *IEEE Transactions on Evolutionary Computation* 23 (2) (2018) 173–187.
- [33] H. Wang, L. Jiao, X. Yao, Twoarch2: An improved two archive algorithm for many-objective optimization, *IEEE Transactions on Evolutionary Computation* 19 (4) (2015) 524–541.
- [34] Y. Liu, D. Gong, J. Sun, Y. Jin, A many-objective evolutionary algorithm using a one-by-one selection strategy, *IEEE Transactions on Cybernetics* 47 (9) (2017) 2689–2702.
- [35] M. Li, S. Yang, X. Liu, Shift-based density estimation for pareto-based algorithms in many-objective optimization, *IEEE Transactions on Evolutionary Computation* 18 (3) (2014) 348–365.
- [36] D. Gong, G. Wang, X. Sun, Y. Han, A set-based genetic algorithm for solving the many-objective optimization problem, *Soft Computing* 19 (6) (2015) 1477–1495.
- [37] Z. He, G. Yen, Many-objective evolutionary algorithm: Objective space reduction + diversity improvement, *IEEE Transactions on Evolutionary Computation* 20 (1) (2016) 145–160.

- [38] H. K. Singh, A. Isaacs, T. Ray, A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems, *IEEE Transactions on Evolutionary Computation* 15 (4) (2011) 539–556.
- [39] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, in: *Evolutionary multiobjective optimization*, Springer, 2005, pp. 105–145.
- [40] S. Huband, P. Hingston, L. Barone, R. L. While, A review of multiobjective test problems and a scalable test problem toolkit, *IEEE Transactions on Evolutionary Computation* 10 (5) (2006) 477–506.
- [41] C. Ran, M. Li, T. Ye, X. Zhang, S. Yang, A benchmark test suite for evolutionary many-objective optimization, *Complex and Intelligent Systems* 3 (1) (2017) 67–81.
- [42] H. Li, K. Deb, Q. Zhang, P. Suganthan, L. Chen, Comparison between moea/d and nsga-iii on a set of novel many and multi-objective benchmark problems with challenging difficulties, *Swarm and Evolutionary Computation* 46 (2019) 104–117.
- [43] R. Shen, J. Zheng, M. Li, J. Zou, Many-objective optimization based on information separation and neighbor punishment selection, *Soft Computing* 21 (5) (2017) 1109–1128.
- [44] X. He, Y. Zhou, Z. Chen, Q. Zhang, Evolutionary many-objective optimization based on dynamical decomposition, *IEEE Transactions on Evolutionary Computation* 23 (3) (2018) 361–375.
- [45] K. Li, A. Fialho, S. Kwong, Q. Zhang, Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation* 18 (1) (2014) 114–130.
- [46] J. Zou, C. Ji, S. Yang, Y. Zhang, J. Zheng, K. Li, A knee-point-based evolutionary algorithm using weighted subpopulation for many-objective optimization, *Swarm and Evolutionary Computation*.

- [47] F.-Q. Gu, H.-L. Liu, A novel weight design in multi-objective evolutionary algorithm, in: 2010 International Conference on Computational Intelligence and Security, IEEE, 2010, pp. 137–141.
- [48] J. Siwei, C. Zhihua, Z. Jie, O. Yew-Soon, Multiobjective optimization by decomposition with pareto-adaptive weight vectors, in: 2011 Seventh International Conference on Natural Computation, Vol. 3, IEEE, 2011, pp. 1260–1264.
- [49] F. Gu, H.-L. Liu, K. C. Tan, A multiobjective evolutionary algorithm using dynamic weight design method, *International Journal of Innovative Computing, Information and Control* 8 (5 (B)) (2012) 3677–3688.
- [50] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach, *IEEE Transactions on Evolutionary Computation* 18 (4) (2014) 602–622.
- [51] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, Moea/d with adaptive weight adjustment, *Evolutionary Computation* 22 (2) (2014) 231–264.
- [52] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, *IEEE Transactions on Evolutionary Computation* 20 (5) (2016) 773–791.
- [53] X. Cai, Z. Mei, Z. Fan, A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors, *IEEE Transactions on Cybernetics* 48 (8) (2018) 1–14.
- [54] M. Li, X. Yao, What weights work for you? adapting weights for any pareto front shape in decomposition-based evolutionary multi-objective optimisation, *arXiv preprint arXiv:1709.02679*.
- [55] L. R. de Farias, P. H. Braga, H. F. Bassani, A. F. Araújo, Moea/d with uniformly randomly adaptive weights, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, ACM, 2018, pp. 641–648.
- [56] J. Li, G. Chen, M. Li, H. Chen, An adaptative reference vector based evolutionary algorithm for many-objective optimization, *IEEE Access* 7 (2019) 80506–80518.

- [57] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Adaptation of scalarizing functions in moea/d: An adaptive scalarizing function-based multiobjective evolutionary algorithm, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2009, pp. 438–452.
- [58] I. Giagkiozis, R. C. Purshouse, P. J. Fleming, Towards understanding the cost of adaptation in decomposition-based optimization algorithms, in: 2013 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 2013, pp. 615–620.
- [59] I. Das, J. E. Dennis, Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems, *Siam Journal on Optimization* 8 (3) (1996) 631–657.
- [60] R. B. Agrawal, K. Deb, K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, *Complex Systems* 9 (3) (2000) 115–148.
- [61] W. Hoffmann, Iterative algorithms for gram-schmidt orthogonalization, *Computing* 41 (4) (1989) 335–348.
- [62] T. Ye, H. Wang, X. Zhang, Y. Jin, Effectiveness and efficiency of non-dominated sorting for evolutionary multi- and many-objective optimization, *Complex and Intelligent Systems* 3 (4) (2017) 247–263.
- [63] Y. Xiang, Y. Zhou, M. Li, Z. Chen, A vector angle based evolutionary algorithm for unconstrained many-objective optimization, *IEEE Transactions on Evolutionary Computation* 21 (1) (2017) 131–152.
- [64] T. Ye, C. Ran, X. Zhang, Y. Jin, Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum], *IEEE Computational Intelligence Magazine* 12 (4) (2017) 73–87.
- [65] R. Shen, J. Zheng, M. Li, A hybrid development platform for evolutionary multi-objective optimization, in: 2015 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2015, pp. 1885–1892.
- [66] Z. He, G. Yen, Many-objective evolutionary algorithms based on coordinated selection strategy, *IEEE Transactions on Evolutionary Computation* 21 (2) (2017) 220–233.

- [67] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, Y.-S. Ong, A new decomposition-based nsga-ii for many-objective optimization, *IEEE transactions on systems, man, and cybernetics: systems* 48 (7) (2017) 1191–1210.
- [68] E. Zitzler, J. Knowles, L. Thiele, Quality assessment of pareto set approximations, in: *Multiobjective Optimization*, Springer, 2008, pp. 373–404.
- [69] H. Ishibuchi, K. Doi, Y. Nojima, On the effect of normalization in moea/d for multi-objective and many-objective optimization, *Complex and Intelligent Systems* 3 (4) (2017) 279–294.